University of Oslo
Department of Informatics

Road Tunnel Entrance Recognition System

A Master Thesis

by

Christoffer Nøkleholm

December 1st 2009

Abstract

Road tunnels are often filled with heavily polluted air, which is dangerous to the driver and passengers. The purpose of a Road Tunnel Entrance Recognition System (RTERS) is to automatically protect the driver and passengers from dangers involved when entering a tunnel. These actions include switching to recirculation of air to avoid “inhaling” pollution from the road tunnel, ensuring that the instrument lights are on and adjusting the headlights. In addition the system can alert the driver of any open windows and remove the dew that might build up on the front window when entering a cold tunnel.

This thesis describes work undertaken on developing and testing image recognition methods for detecting road tunnel entrances to be a part of a future RTERS. The system takes a color image as input, evaluate the image and decides if the image contains a road tunnel entrance or not. These methods utilize different features in road tunnel entrances and the best methods are combined to create a more effective hybrid method.

The methods in the RTERS are:

- Tunnel Arch Detection (Circle Hough Transform is used to detect the tunnel arch).
- Consecutive Tunnel Lights (Searches for lanes of tunnel light objects).
- Elliptic Black Hole (Marks the tunnel entrance with the help of Lane Tracking and evaluates the dark tunnel entrance shape).
- Template Matching (Checks the image for matches to a template of a general tunnel entrance).

From these methods, two hybrids were developed:
Hybrid 1: A combination of Tunnel Arch Detection and Consecutive Tunnel Lights
Hybrid 2: A combination of Elliptic Black Hole and Consecutive Tunnel Lights

The final choice of method for the RTERS was Hybrid 2. This program gives an average of 82% correct classification on the two test sets used in the thesis. Hybrid 2 also has an average run-time of 1.4 seconds, which is short enough for the method to evaluate two images in the Photo Capturing Zone. This is the zone that the methods are designed to work within and encompasses the distance between 120 meters to 40 meters from the tunnel entrance.

Foreword

This report from the development of a Road Tunnel Entrance Recognition System is the result of a master thesis at the Department of Informatics, University of Oslo. The thesis was finished in December 2009.

The RTERS can be used in combination with other Intelligent Vehicle Technologies to automatically prepare the driver and vehicle when the vehicle is approaching tunnel entrances. This signal could be used to change the ventilation to recirculation or filtration and turn on the instrument lightning. It could also advice the driver to close all windows.

The project has been a challenge in many ways. It was a struggle to find literature that was related to this specific problem. It was first in June that I found a similar project in Germany and by then I had figured out, by myself, what kind of methods could be used in this kind of system.

During the development of the methods, I had learned how difficult it was to find mutual characteristic for several tunnel entrance designs and methods that could be used to detect these.

My counselor, Professor Jim Tørresen, has helped my greatly in the periods where the work moved slowly. I wish to thank him for all the good advice he gave me through the whole work phase.

I also need to thank my father for all the help he has provided with discussions of problems and giving me new ideas I could explore in depth. He has also helped by getting me back on track when I dwelled on a specific problem for too long.

Last but not least I want to thank the rest of my family and my friends for the support they have given me throughout the process.

Contents

Chapter 1

Introduction

1.1 The objectives

Even though the number of traffic accidents in tunnels is relatively low compare with accidents on open roads, these accidents have often resulted in large headlines. It is not only due to the accident itself, but also the danger the accident represents to other vehicles coming into the dark and narrow tunnel and that it is complicated to perform rescue operations in such situations. This has also made it an important issue on the agenda for public road safety.

The objective of this thesis is to develop a Road Tunnel Recognition System (RTERS) to be included as part of Integrated Vehicle Technologies (IVT). The RTERS shall help the driver to focus on the road and other vehicles when driving through a tunnel. When the RTERS have identified a tunnel entrance it will give a signal to other programs in the IVT that ensure the headlights and instrument lights are on. The dew that might build up inside the front window when entering a cold tunnel could automatically be removed. Additionally the ventilation system will be switched to recirculation to avoid "inhaling" polluted air from the road tunnel. In this way it is avoided that the driver is distracted when he manually is switching on and off these systems.

This project aims on identifying different algorithms which could be used to recognize road tunnel entrances. Each algorithm will be tested on images with and without a tunnel opening to see how exact it is on recognizing different features which a tunnel opening can have. After the algorithms have been run separately, the results will be analyzed to see which would give the best results in the final RTERS. It will be of importance to find the best combination of algorithms which could be integrated in a IVT system.

1.2 Traffic development

1.2.1 From track to road

When the first steam driven vehicles were used for transport of personnel and goods, the roads were built for wagons pulled by horses and oxes. The roads had been continuously developed by the users – from the first track made by wildlife, then by people that used the same track when hunting the animals. They also used these tracks when they walked from one

area to another – either for new hunting grounds or they needed to trade their products with people in other places. This traffic would then increase when more people used the same track. When they started to use horses and oxes for transport, or they needed to move cattle between farms or into towns, there were no reasons to make new tracks. Only in the places where it was too steep for the animals to climb or not suitable for wagons, they needed to find alternative tracks. Even today we have roads in the countryside that obviously follows the same track as they were formed by this early traffic. These roads are sometimes only suitable for slow traffic that can accept narrow, bumpy and curved roads.

During the first years after the motorized vehicles were introduced, the main concern was to protect the existing traffic from the danger caused by the new technology. The new danger was first of all a result of the change from the relatively slow traffic to higher speed, increased noise and heavy vehicles. The minor traffic regulation and control could not match the constantly changing traffic situation.

When the traffic increased in number and speed, the accidents increased in both number and seriousness. Because of this, safety issues were moved higher and higher up on the agenda. Authorities were constantly looking for measures that could make it safe to travel on the public roads.

1.2.2 Road tunnels

Very early in the process of increasing the road safety one has built road tunnels. Road tunnels are made to avoid dangerous narrow roads that had to be built like shelves in the mountainside. These roads are exposed to possible avalanches and the steep roadsides are dangerous if the vehicle should accidently slide off the road. During wintertime the mountain roads could be closed for safety reasons due to heavy snowfall.

The tunnels in urban areas are built to let the traffic pass cities underground due to lack of space for the increasing number of motorized vehicles and to reduce the traffic noise in city centers.

All these tunnels are increasing in numbers, length and dimensions year by year. Even though the tunnels are built to avoid dangers, they represent new safety challenges. The first potential danger is the change from the road in an open area to the limited space in a tunnel. Inside the

tunnel it is the polluted air and the darkness which represents risk factors to the driver and passengers.

There are other safety challenges with road tunnels, such as illumination, warning signs, lane separation and safe exit if an accident happens inside the tunnel, but this will have influence on how the tunnel should be designed.

1.3 Tunnel traffic

1.3.1 The main challenges

According to "Trafikksikkerhetshåndboken" published by Transportøkonomisk Institutt [Transp00] (a Norwegian institute for transport cost-benefit analysis), which compare the number and seriousness of accidents on different types of roads and tunnels, the accidents will be reduced when roads are replaced by tunnels. Not as much for motorways and countryside roads as with urban roads with a mix of vehicles, people on bicycles and pedestrians. In these areas where all types of traffic are close or even cross each other's "track", the risk of accidents are higher than if the vehicles are driving in tunnels separated from the "soft" traffic and without all the crossing roads you will have in a city.

The TI report gives the following list of challenges with tunnels:

1. The tunnel gives limited space for avoiding maneuvers in critical situations.
2. The daylight is shut out and the light contrast is strong at both the entrance and the exit.
3. The air is polluted, plus reduced sight due to moisture and exhaust.
4. Accidents may block exits and make access difficult for emergency assistances.

1.3.2 High-speed hazards

The TI report did not include any analysis of the effect of installing continuous camera surveillance of the traffic in tunnels, as they had no documentation on how this would influence the number and seriousness of accidents. They refer however to experiences from the first months after opening of a tunnel in the city of Oslo (Vålereng-tunnel). Due to the design of curves combined with steep angle (approx 6%) inside the tunnel there were several accidents. After installing an automatic speed-monitoring camera combined with high penalties to those who did not respect the signs for reduced speed, the number of accidents

was reduced significantly. It was obvious that the unchanged high speed from the open road into the tunnel caused accidents.

1.3.3 Shifting light conditions

The change from the road in a wide-open and bright area to the darkness of a tunnel is one of the main dangers connected to the road tunnel system. It is widely accepted that the tunnel designers should focus on how to reduce this contrast as much as possible.

One of the most effective details that have been an accepted standard is installation of much light in the tunnel openings. This is gradually reduced to the level which gives sufficient light for driving through the tunnel. At the end of the tunnel, the light intensity is increased again to reduce the contrast from the darkness of the tunnel to the daylight outside.

With a vehicle-integrated system that detects tunnel entrances, the driver could be warned about this danger. The problem could also be lessened with advanced controlled lighting in the car that would prepare the drivers eyes to the coming light change. This technology does not exist at the moment but with a road tunnel entrance recognition system, the needed foundation for a technology that lessens the light change is available.

1.3.4 Severe air pollution in road tunnels

The obvious drawback by having the traffic in a tunnel – specially a long one – is that the pollution from the traffic will represent a high-risk danger to the people that drive through the tunnel.

A study, which has been published by Queensland University of Technology [Morawska09], measured ultra fine particle concentration levels outside a vehicle traveling through the M5 East tunnel in Sydney. The study concluded that road tunnels were locations where maximum exposure to dangerous ultra fine particles in addition to other pollutants occurred.

The study aimed to identify the concentration levels found in the tunnel. It generated a huge body of data on the concentrations and the results show that, at times, the levels are up to 1000 times higher than in urban ambient conditions.

This study confirms that the health risk one will be exposed to by traveling through tunnels is so severe that measures have to be taken. The ventilation system within the tunnel must be able to change the air sufficiently. The drivers must close car window and switch to recirculated air.

These actions could done automatically by the use of signals from the RTERS, and let the driver focus on the road instead of looking down to press buttons that perform these tasks.

1.4 Traffic safety technologies

1.4.1 Automatic traffic control systems

The computer era gave us the tools to go from manual traffic control to a more cost reductive and more efficient automatic traffic control. With the use of traffic signals in the big cities, it is possible to adapt the signals to handle different traffic situation. During the rush hours in the morning the traffic needs high capacity on the roads into the city and out of town in the afternoon. The traffic control centre may then change driving direction on one or several lanes to adapt the road system to the actual situation, though this is not widely used in Norway. The so-called “green wave” can also be adjusted to ensure the traffic runs more smoothly.

Lately computer science has provided a more automatic solution to traffic control, which might remove human surveillance completely. This technology is called Intelligent Transport Systems (ITS) and is making huge advances in the struggle to minimize traffic accidents and improve traffic flow. An example for this is a system that collects video surveillance of the highway, which is analyzed by programs that check the continuous flow of information and react automatically when unwanted situations occur. Such typical situations would be traffic jams during rush hours or a standstill caused by traffic accidents. Then the ITS can close the access to the area and bypass the traffic until the situation has been cleared.

In traffic accidents the ITS can automatically alert the traffic control center which will send ambulances, police and the fire brigade to the place of accident.

1.4.2 Automatic safety systems in vehicles

Several projects aims on developing stationary systems for automatic detection of traffic situations such as described above, but there are also such systems integrated in vehicles. This kind of technology is called Intelligent Vehicle Technology (IVT).

There are detection systems that give signals when the vehicle comes too near an object behind the vehicle when parking and when the distance to the vehicle driving in front is too short to be safe. Some cars have surveillance camera that detects speed signs along the road and warn the driver if the speed is too high.

Navigation systems (GPS) installed in vehicles could warn when the vehicle approaches a preprogrammed danger zone; such as an approaching tunnel or if the vehicle is too close or in, the lane for meeting traffic. The limitation of such system is the lack of automatic online updating when the navigation system loose contact with the satellites. This could be compensated if the system calculates the position by the use of information from the trip counter.

A vehicle-integrated system could also monitor the general health of the driver – and stop the car if the situations become dangerous. If driver is tired, drunk or not able to drive due to drugs, then system can then make it impossible to start the engine.

An automated detection of lane markings can discover that the vehicle is headed out from the correct lane and alert the driver. This can prevent the driver from colliding with other vehicles or driving of the road, either due to falling asleep or being temporary distracted.

1.5 Pattern recognition

Pattern recognition is used in many applications; from satellite surveillance to facial recognition and product quality control on assembly lines. The main cause for the development of pattern recognition systems is to reduce the need of human supervision. A system that functions without human supervision is cheaper, possibly more accurate and can be constantly active without any brake as long as needed. This system will however not be able to handle unexpected situations as a human could, with ability to analyze and act accordingly.

A pattern recognition system works perfectly as an aid to a human. The RTERS is a typical sample of such system, as it will assist the driver by initiating preprogrammed supportive actions while leaving the driver with full control of the vehicle.

The pattern recognition system analyses an image by the use of different types of methods. The first process that is executed is segmentation, which finds regions of interest by looking at the pixel values in the image and the pixel value difference between neighboring pixels. Secondly the sought shape or pattern is search for in the regions of interest found by the segmentation. If any shapes or patterns are found then these are analyzed and classified as either the sought object or an unwanted object.

For this RTERS the pattern recognition system will search for lines, curves and contrasts which are typical for this type of constructions. If any such patterns are detected, they are analyzed further to reach an accurate classification. This process is described in chapter 2.

1.6 Structure of the report

This report is divided into 6 chapters. The first chapter gives an ***Introduction*** to the motivations behind the thesis and introduces pattern recognition as a tool for many applications.

The second chapter, ***Background***, gives a summary of what has been done to improve the safety on the roads and how these measures have developed with the expanding traffic. Then follows a presentation of pattern recognition theory used in this thesis and references to works similar to the topic introduced.

The third chapter, ***Road tunnel entrance recognition methods***, starts with a look at the features in tunnel design and describe every method that has been tested in this thesis.

Chapter four, ***The hybrid system***, analyzes the implemented methods and describes hybrid systems that can enhance the performance.

Chapter five, ***Further work,*** suggests what could be the next steps to improve and develop the RTERS presented in this thesis further.

The last chapter, ***Conclusion***, gives a summary of the preparation for this work and the progress of the RTERS development with tests and result evaluations.

Chapter 2

Background

This chapter will present different methods in image analysis and explain why one approach has been chosen before another. Previous work is described in the last paragraphs of this chapter.

Pattern recognition is often divided into 3 parts. The first is the segmentation, to remove unwanted information and reveal wanted information. Then there is localization of the sought object in the image. And lastly there is object analysis. Each of these steps contains choices one has to take and the effectivity of the system greatly depends upon making the correct ones. Each decision has therefore been closely examined and evaluated to make the correct choice.

2.1 Color representation

Before starting the pattern recognition process, it is important to establish a color representation. There are three main color representations that are relevant for the RTERS. RGB is the first and most widely known representation. It uses three primary colors to make up the RGB color space, Red, Green and Blue. This color model is in color TVs, computer displays, video cameras and digital photo cameras.

CMY is the second representation and stands for cyan, magenta and yellow. This representation is mostly used in color printing. This is because CMY is a subtractive color model which means that white is the absent color and black is the combination of every color. RGB is the opposite of this; it is an additive color model where white is the combination of every color and black is the absence (Figure 2.1). Because these models are opposites, the conversion between them is very simple. The tight bond between the models mean that there is not any big difference in color manipulation gains from either.

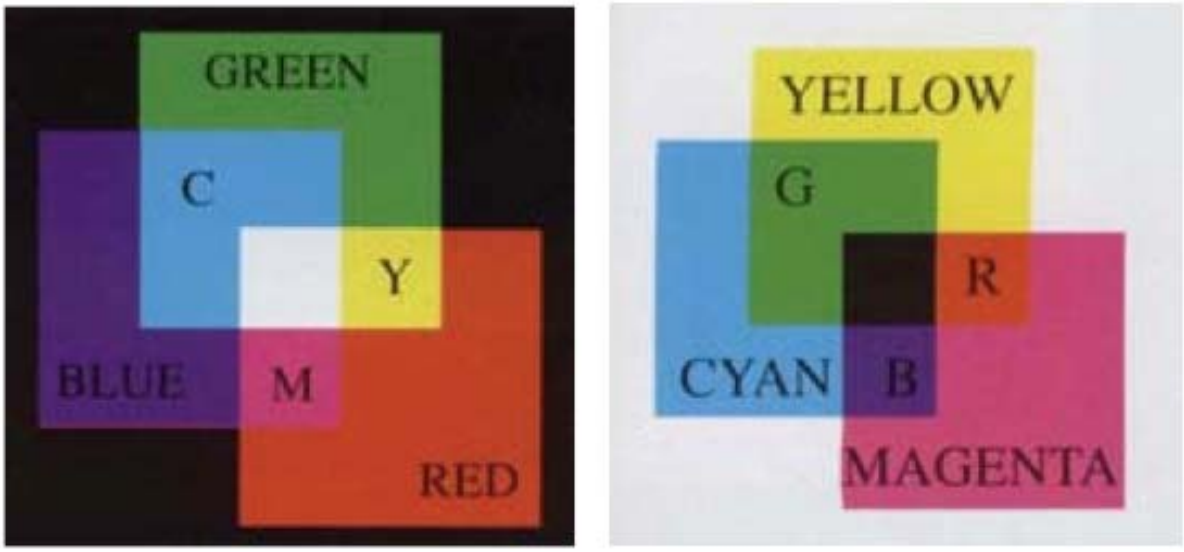

Figure 2.1: RGB (left) and CMY (right) [Foley97]

The last representation is the HSI model and is the model that best compares to how humans see color. HSI do not have three primary colors that are combined to make one specific color but describes a color from its Hue, Saturation and Intensity. Hue represents the pure color. Saturation represents the degree the pure color is "watered down". Intensity is the gray-level or the brightness of the color. HSI is often used color descriptors to make it easier for humans but it isn't ideal for color generation in hardware.

The question here is if it's better to use color representation that uses primary colors as bases or a more intuitive representation. For the reason that cameras mostly uses the RGB model and because its advantage in segmentation (which is explained in 2.2) the choice falls on RGB.

2.2 Segmentation

Segmentation is the process of dividing the image into regions of importance. Segmentation of an image usually captures one specific part of the image by elevating the interesting regions and removing uninteresting data. A region is a collection of pixels with similar characteristics, such as color, intensity or texture. When applied to a nontrivial image, segmentation is one of the most difficult tasks in image processing. Trying to segment a wide range of tunnel shapes from the surroundings can be classified as a nontrivial process.

Segmentation can be divided into two categories, region and edge based methods. The difference between these two is that region based segmentation looks at similarity in the intensity values and edge based segmentation looks at discontinuity in the intensity values. One category defines the interior of an object, while the other defines its border. These representations are, in a way, similar since you can find either one with the other.

The region based segmentation methods that will be explained here are thresholding and region growing. There exists many edge based segmentation methods. Some of the more important will be explained in chapter 2.2.3.

2.2.1 Thresholding

Thresholding is a technique for partitioning an image directly into regions based on intensity values. This is probably the simplest method, but it holds a central position in image segmentation, because of its simplicity in implementation and computational speed. There are multiple ways to do thresholding. Global thresholding takes a constant threshold T and applies it for every pixel in the whole picture, if the intensity is higher than T then it is set to 1, if it is lower than or equal to T then it is set to 0.

In variable thresholding, however, the threshold varies according to the properties of the neighboring pixels of each point in the image; this makes it less affected by variances in global luminosity. Global luminosity variances are usually not part of the sought information in images and it is for that reason variable thresholding often is chosen over global thresholding.

Both these techniques are used on a single variable, gray-scale intensity. This is an effective simplification if the sought information is easily distinguishable by intensity, but for color images it might be wise to apply the thresholds to multiple variables. When the wanted regions have a distinct color this is especially relevant. As mentioned in chapter 2.1 there are more than one way of representing colors, and for that reason, multiple ways to color threshold. HSI is probably the most natural color space to think of, because of how suitably color is represented in the hue specter. Even though this is intuitively correct, segmentation generally gives better results when RGB color vectors are used [Gonzalez92].

In this thesis there are areas where thresholding can make the localization process easier. When looking for a dark part in the image (the tunnel entrance), global thresholding can be applied to capture the low intensity pixels and remove high intensity pixels. Doing this will make it easier to further evaluate the interesting regions in the image. Color filtering can also help in locating tunnel lights, which is done in the Consecutive Tunnel Lights method [chapter 3.3.2].

2.2.2 Region-growing

Region-growing is based on the idea that neighboring pixels, in the area of interest, have similar intensity values. The method starts out with a set of seeds (pixels), which marks the objects to segment, and expands these regions by comparing the pixels to neighboring pixels. Neighboring pixels with little difference in the intensity are added to the region. This way the regions grow recursively until there are no more similar pixels at the region border and all pixels are included into a region.

Because of the methods high dependency on the placement of seeds, the methods efficiency relies heavily on feeding the method with seeds inside the sought regions and that those seeds are undisturbed by noise in the image.

Region-growing is tested as an alternative to Thresholding in the Elliptic Black Hole method [chapter 3.3.4].

2.2.3 Edge-based segmentation

The edge or border of an object in an image is defined by changes in the intensity value between neighboring pixels. The result is displayed in a binary edge-image and is usually derived from a grayscale image. It is possible to create edge-images from color images also, but it is easier if the color image is changed into a grayscale image first.

To test if neighboring pixels has a significant difference in intensity values, different edge-based segmentation methods use different smoothing filters. These smoothing filters calculate the gradient of the image intensity for each pixel. The result from this is the direction of the increase in intensity in the image and the rate of change in that direction. This information can then be used to find the likelihood of an edge in each pixel and from this create a binary edge-image.

The Sobel operator, the Prewitt operator and the Roberts' Cross operator convolve two filters with the image. One of the filters is for finding horizontal edges and the other is for finding vertical edges. Equation 2.1, Equation 2.2 and Equation 2.3 show the different filters when applying them to the image.

"If we define A as the source image, and G_x and G_y are two images which at each point contain the horizontal and vertical derivative approximations, the computations are as follows:" [WikiSobel]

$$\mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A}$$

Equation 2.1: Calculation of the derivative approximations with the Sobel operator [WikiSobel]

$$\mathbf{G_x} = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G_y} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix} * \mathbf{A}$$

Equation 2.2: Calculation of the derivative approximations with the Prewitt operator [WikiPrewitt]

$$\mathbf{G_x} = \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G_y} = \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix} * \mathbf{A}$$

Equation 2.3: Calculation of the derivative approximations with the Roberts' Cross operator

Two other edge-based segmentation methods have also been tested in this thesis, the Laplacian of Gaussian method and the Canny method. The Laplacian of Gaussian method detects edges by looking for zero-crossings after filtering with a Laplacian of Gaussian filter. Zero-crossings are zero values in the second derivative and represent an alternative method from looking for maxima and minima in the first derivative, which the previous three methods used.

The Canny method uses a gradient filter calculated from the derivative of a Gaussian filter. It uses two thresholds to detect both strong and weak edges. The weak edges are combined with the strong edges in the output if they are linked to strong edges. This approach is less affected by noise in the image.

All these edge-based segmentation methods are implemented in Matlab Image Processing Toolbox, and have been tested in this system without modifications. Examples of edge-segmentation on a tunnel image and a plain road image is shown in Image 2.1 and Image 2.2

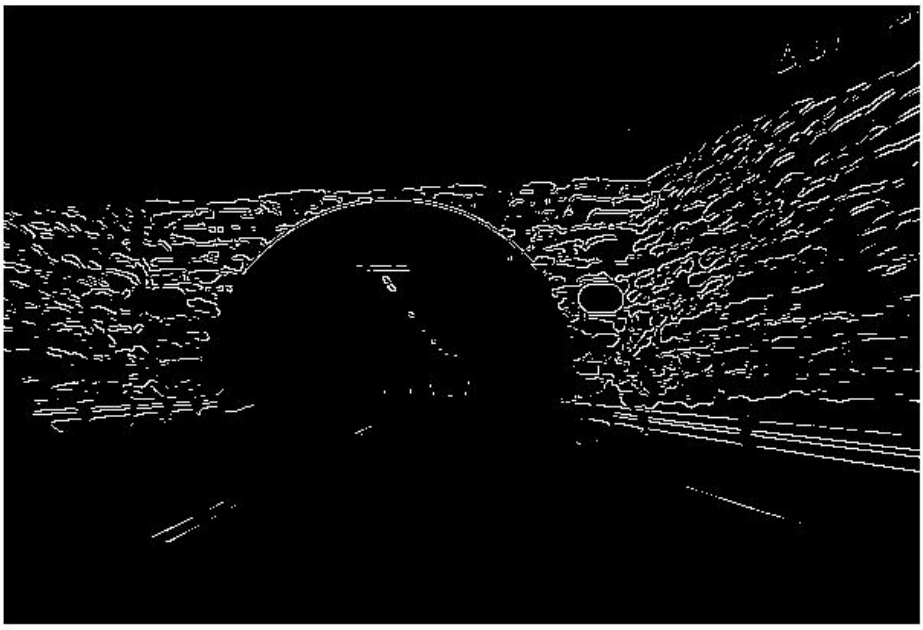

Image 2.1: Edge-image of a tunnel (with Roberts' Cross)

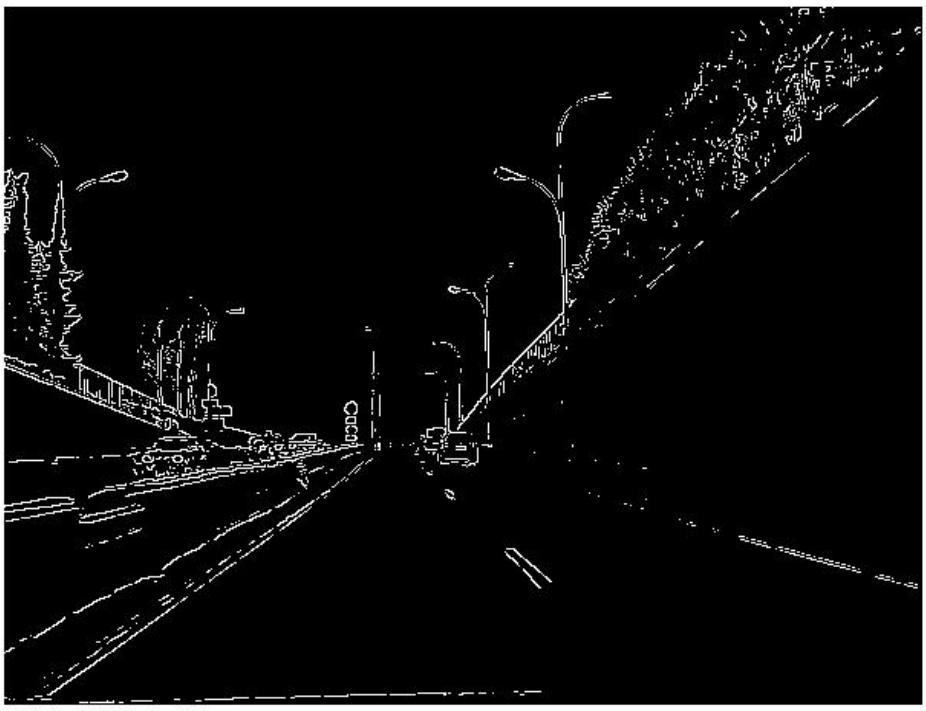

Image 2.2: Edge-image of a road (with Roberts' Cross)

An edge-image can be very useful if the sought object has clear boundaries (separates itself from the background). Getting a good result is difficult when working with complex shapes or images filled with texture. But even if the result is not optimal the edge-image can contain valuable information, which can be extracted with a good algorithm. One way to use an edge-image is the Hough Transform [chapter 2.3.1].

2.3 Localization

2.3.1 Hough Transform

Hough transform is a special method to find shapes in pictures. The method is often used to find straight lines and is for that reason repeatedly used with Lane Tracking [chapter 3.3.3]. Even though the method is mostly used to find straight lines, the method can be adapted to find multiple shapes that can be described by a finite set of parameters. This can for example be circular or half-circular shapes as used in the TAD method [chapter 3.3.1].

If we look at line detection; the method goes through every edge-pixel in the edge-image [chapter 2.2.3] and registers the corresponding line for every pixel in the parameter space. A line in the original image can be described with the equation $y = ax+b$. This line can be represented by a point in the parameter space with the equation $b = -xa+y$ (here a and b are

the variables, while x and y are constants). This means that every point in the image can be represented as a line in the parameter space (and vice versa). In the Hough Transform the parameter space is represented as the accumulator space. This is because the values in the parameter space accumulate when multiple lines cross each other. An example of line detection is shown in Image 2.3.

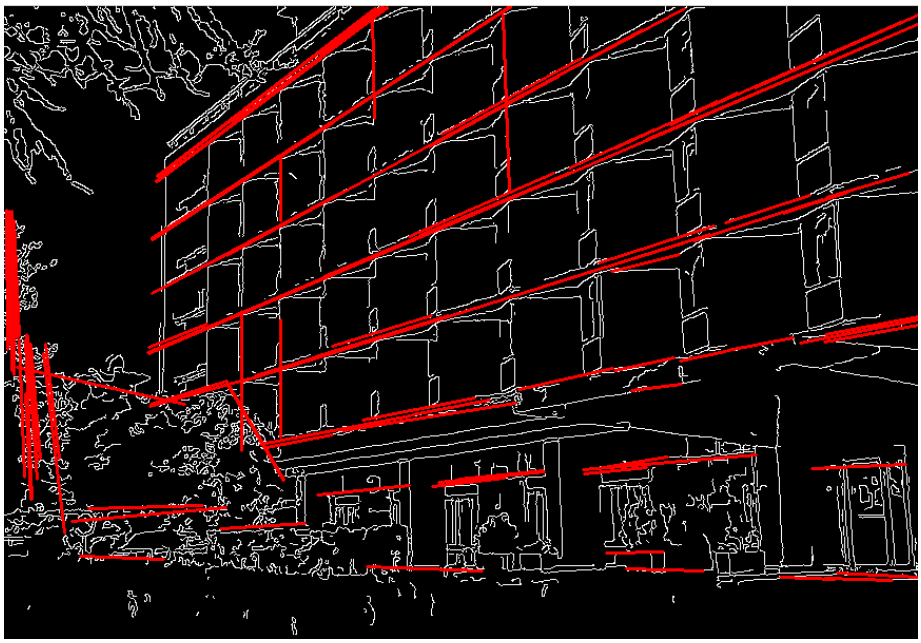

Image 2.3: Linear Hough Transform [houghp]

When the Hough Transform has mapped each point in the image into a line in the accumulator space, the highest values represent the most clearly stretched lines in the original image. With this information, every line above a certain threshold is accepted as straight lines in the image and can for example be use to mark lanes in a Lane Tracking method.

2.3.2 Template Matching

Template Matching is a method for finding specific objects or patterns in images. Templates are small images of the objects that one wants to find in larger images. Similarity to the template is tested for in the whole image. If the templates similarity is above a set percentage then the method can conclude that the sought object is in the picture and get the coordinates to that object.

The method is most basically done by convoluting, which is similar to cross-correlation, the template and the image. This is done by going through every pixel in the image and calculating the total of products between the coefficients in the template and the image for every pixel in the template. The highest total will then mark the best fit.

An alternative to this approach is to filter in the frequency domain instead. If the template is above a certain size then this alternative is faster than the previous. The frequency domain of an image describes the image by its frequencies and phases. If two images match in pixel values then their frequencies and phases will also match. This approach is used in the Template Matching in this thesis [chapter 3.3.5]. The algorithm used can be explained in 6 steps:

1. Padding the template.
2. Performing Fourier transform of template.
3. Performing Fourier transform of image.
4. Multiply the result from 2 and 3.
5. Performing Inverse Fourier Transform on 4.
6. Find the best match

2.4 Object analysis

When an object has been localized, the next step is to check if the object matches certain descriptors taken from the sought object. It is important to choose descriptors or features that best describe what is special about the sought object. These can be contour descriptors like area size, perimeter length and curvature, topological descriptors like number of holes in the object, number of connected components and symmetry, or more exotic descriptors like moments taken from physics and statistics.

Matlab Image Processing Toolbox includes the functions; regionprops() and bwlabel(), which make the object analysis process much easier. The function bwlabel() goes through an image where objects are white and the background is black, and marks each object region with a numbered pixel value. This means that the label image that bwlabel() returns has the objects marked with different numbers. This label image is then given to regionprops() so that it can extract object descriptors from each labeled object. These object descriptors include:

1 Area (the number of pixels in the object)
2 BoundingBox (upper left coordinates and length and width)
3 ConvexHull (the smallest convex polygon that can enclose the object)
4 Eccentricity (how circular the object is; 0 if a line and 1 if fully circular)
5 Orientation (the angle between the x-axis and objects major)
6 Perimeter (the length of the objects boundary)

After extracting one or more of these descriptors from objects in an image set, they can be visualized with gscatter(). This constructs a two-dimensional plot of the distribution of objects in between two descriptors. If the plot shows groupings of sought objects then the descriptors are ideal. An example of this, with number recognition, is shown in Figure 2.2.

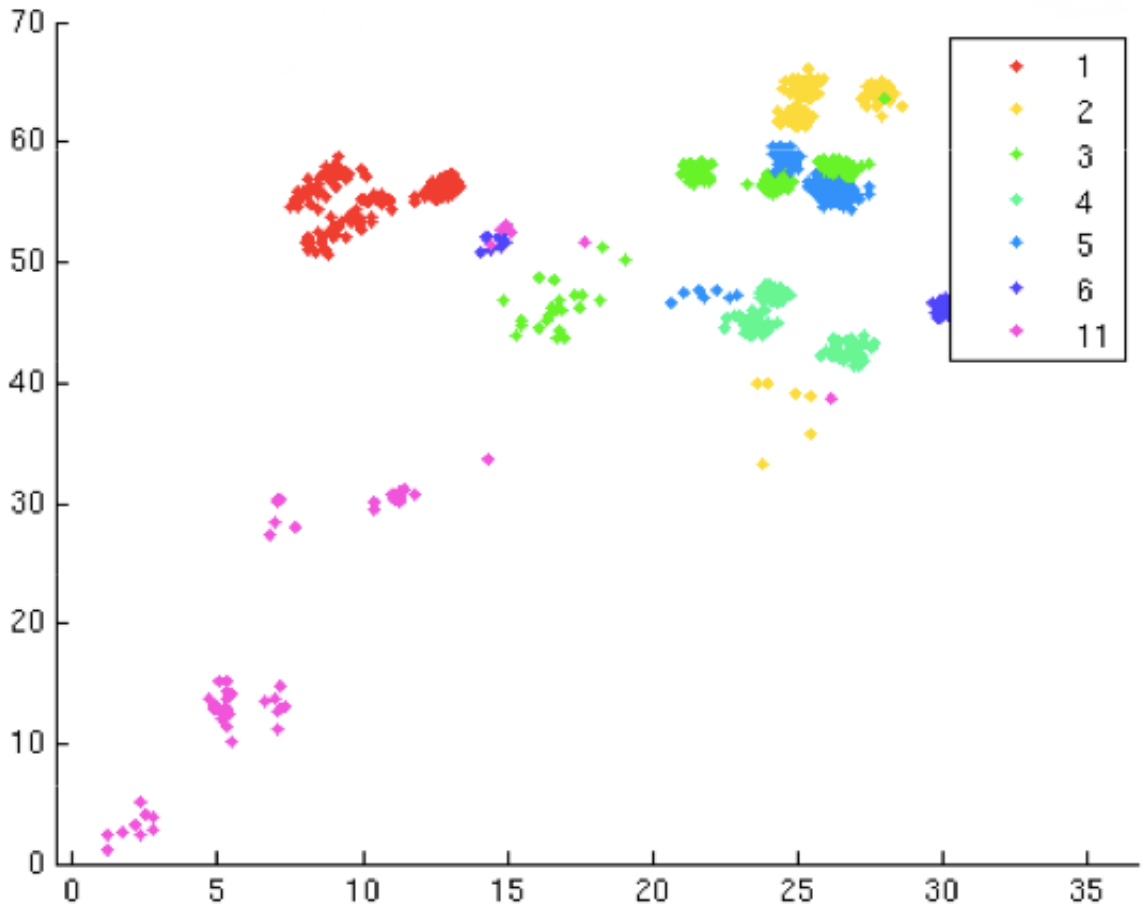

Figure 2.2: MinorAxisLength vs. MajorAxisLength for Symbols 1-6 and outliers (11) [Solberg08].

2.5 Previous work

When the objectives of this master thesis were established there were not found any references to similar work that focused on tunnel entrance recognition. Later in the process of developing the entrance recognition system it was discovered a reference to such work in a German technical article regarding video processing in Driver Assistance Systems.

There are several works on road sign recognition and lane tracking available. The methods used and angles to solve the challenges in this thesis were based on information collected from several such scientific articles. The articles [Tørresen04], [Jamal] and [Nagumo] have been central in the preparations for this work. These articles presents IVT with focus on camera applications with the camera directed towards the environments and the traffic in front of the vehicle. No other article relevant to the topic of this thesis, Road Tunnel Entrance Recognition System, has been found.

Two of the articles [Tørresen04] and [Clause06] will be presented here to introduce previous work in this field.

2.5.1 Efficient Recognition of Speed Limit Signs [Tørresen04]

The article describes the research on detection of Norwegian speed limit signs by the use of a camera installed on vehicle. This represents similar works perfectly because it gives a systematic introduction to how objects can be recognized on photo images.

The speed limit sign recognition system use a standard three-step method to detect and recognize the speed limit signs:

First step was to make a color filtration to expose the red circle on the speed limit sign. This could be done due to the exclusion of End of Speed Limit signs that do not have this circle. They use image-filtering RWB to separate the color red plus white and black from the picture. This filtration uses the RGB color model with a wide specter to ensure a robust system. The light intensity of the images is fixed by adding the difference between the average pixel value and a given limit value in the RWB picture.

Second step was to use Template Matching to locate the signs. It was used 6 templates to locate the red circle on the signs. In addition the algorithm for template matching removes the objects that are not speed limit signs to make the process more effective.

The third step was to use number recognition to identify the speed limitation given within the circle. This is done by first removing useless information, followed by finding the edge round the numbers, excluding the second number (as it will always be a zero), converting the figure to a 7x5 bit array and classifying the bit array. The method for classification of figures is a feed-forward neural network computed by the back-propagation algorithm.

When the algorithm was tested they used 196 images and 115 of these included speed limit signs while the remaining 83 showed other types of signs – or even without any signs.
The test images represented a wide variation of difficulty levels to test several possible situations. The test result was 87% correct recognition of images with a speed limit sign. The actual speed limitations on these signs were all correctly classified. 94% of the images without speed limit sign were sorted out, the rest were mistaken as speed limit signs but corrected in the last part of the algorithm to 96%. Totally were 90,9% correctly classified. These results were made at an average processing speed of 130ms per image – or approx 8 images per second (by the use of a 1,4 GHz Athlon AMD processor).

2.5.2 Tunnel Entrance Recognition for video-based Driver Assistance System [Clause06]

Half way through the work with this thesis it was discovered a reference to one previously published article about road tunnel entrance recognition. The reference was found in the presentation of *"Using Partial-Run-Time Reconfigurable Hardware to accelerate Video Processing in Driver Assistance System"* written by Dipl.-Ing. Christopher Claus, Technical University, Munich, Germany - who also stood behind the reference article.

The most significant difference between the system presented in this thesis and this article [Clause06] is the approach to the problem.

The German tunnel entrance recognition system has been tested on two video streams from a vehicle approaching tunnel entrance on a German highway. The tunnel entrance geometry and tunnel illuminations in the test videos are probably of the same type and design. How it would function with tunnel entrances of different design is not presented.

The first sequence of the system is that it looks for tunnel entrances shown in a stream of images from a video camera. When the next tunnel entrance image has been identified and it is bigger than in the previous image, then the Region of Interest (ROI) is marked and saved in a buffer.

The tunnel entrance is identified in a six-step program.

Step 1	First the system uses an Edge Filter to identify a possible tunnel entrance.
Step 2	Structuring lines. Lines on a tunnel entrance would not be "zigzagged or too short". The lines will be curves or lines with a minimum length, which in addition have only negative or positive grades.
Step 3	Lane Detection by the use of Radon Transformation. This is to have it confirmed it is a tunnel entrance.
Step 4	If there is a dark area between the lines it is supported to be a tunnel entrance.
Step 5	If the second tunnel entrance found is larger than the previous found and saved in the buffer, it is interpret to be a situation that the vehicle is approaching a tunnel entrance.
Step 6	If a Motion Estimation of the moving vehicle could confirm that the size and direction of the first entrance would be equal to the second image due to the speed and driving direction of the vehicle – it is confirmed to be a tunnel entrance.

The German Tunnel Entrance Recognition System is described in the following flowchart:

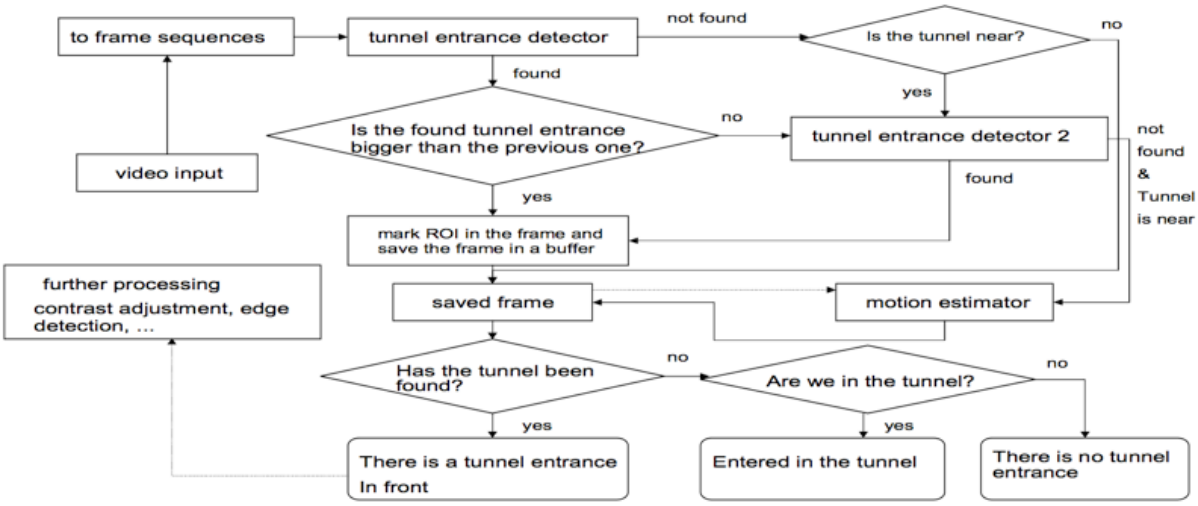

Figure 2.3: Flowchart of [Clause06]

2.5.3 Previous work compared with this work

Only the first report referred above [Tørresen04] has had an influence on the work with this thesis. The structure of the report and how it approaches this type of problem have been helpful. The method used for recognition of speed limit signs (Template Matching) was included as a method that had to be tested in this thesis. It was clear though, from the start, that there are distinct differences between a speed sign and a tunnel entrance as objects. The speed sign will have the same characteristics (color, shape and size) all the time while tunnel entrances comes in several shapes. Even the contrast to the surroundings would have a great influence on which method or which combination of methods should be used in a tunnel entrance recognition system.

As described in chapter 2.5.2, the project [Clause06] aimed also on recognition of tunnel entrances but was based on analyzing a stream of video images. This gave the project a different angle to the problem as it could compare images that followed each other and from that conclude if the vehicle was approaching a tunnel entrance or not. The limitation of this system seems to be that it is developed to recognizing tunnel entrances of German motorway standard. These “clean cut” and repeatedly found shapes are rarely found in the Norwegian road net.

This thesis will base the development of a RTERS on three sets of still photos and each of these series will include images taken of different tunnel entrances. There will not be any link

between the images in these image sets to be analyzed as there are in [Clause06]. This thesis will focus on analyzing each photo separately.

Chapter 3

Road tunnel entrance recognition methods

In this chapter, different tested methods are described and the reasoning behind them is explained. Some methods come from other science papers; these will be described in relation to how they work in their original environment and then how they are adapted to this problem. Most of the methods originate from ideas created by analyzing the problem and studying faults in other methods.

After each method has been presented, a more thorough description follows together with how the method was implemented. In this part, the ideas are put to practice and their worth tested. Every method is tested as if being the single method the system will contain. This leads to a simplified evaluation of the methods and may not be how the methods will be used in the final version of the program.

3.1 Preparations

The first step that had to be done, beyond studying earlier work, was to collect images for the experiments. This was undertaken by driving around in Oslo and Østfold and taking pictures with a Canon IXUS 800IS camera. The pictures were taken from the passenger site, out through the front window. In some of the images, the camera was too far from the front window, which caused reflections in the window to be clearer. When a real ITS-camera is mounted, this should not be as big problem as in the photos taken here.

The pictures were taken to obtain a wide range of potential road tunnel images. This photo session was done in January in Norway, which meant that all the tunnels were partly covered in snow. Because of this there was a need for a second “photo shoot” when the snow had melted. This “photo shoot” was executed in April but this time with video cameras. A Canon HF10 camera and a camera loaned from the University of Oslo developed by Micron Imaging were used this time. With the Micron Imaging camera (RoadRunnerDemo), it was possible to get the video streamed directly to the computer as it would during a real-time prototype test of the system. A third “photo shoot” were done in the summer while traveling over the Norwegian mountains. Later in the thesis, these images proved to be a bit different from the rest, which will be explained in chapter 4.1.

Through the three “photo shoots” hundreds of images were gathered that could be used to test the developed methods. These images were divided into 2 test groups: the training set and the test set. The training set was used to optimize the results, while the test set were just to test the optimization on the training set. In each set, there are 100 images. 50 of these are of tunnel entrances and the other 50 are without tunnel entrances.

Because of a spread in results using the training set and the test set, a third set was made October 31. Since this was done late in the research process, not every test could be repeated with this second test set. The test set was, however, run on the end results from every method to give a better summary of how effective the methods were. This test set has 100 tunnel images and 100 non-tunnel images.

The tunnel images are a mix of circular tunnels and square tunnels taken from Photo Capturing Zone (see Figure 3.1). (There are fewer square tunnels in real life and therefore also in these sets). The Photo Capturing Zone have been chosen after evaluating the amount of time needed to check for tunnels and the difficulty of evaluating images taken from a larger distance. The zone is meant as an approximation used for manual data collecting. In Table 3.1 you can see that the shortest time it takes to drive 40 meters is 1,4 seconds and the longest time it takes to drive 120 meters is 7,2 seconds. 1,4 seconds is set as a preliminary minimum time limit for the recognition system. Images taken from a greater distance than 120 is unnecessary because 4,3 seconds is a long enough time limit for the recognition system.

	40 meters	**80 meters**	**120 meters**
60 km/h	2,4 seconds	4,8 seconds	7,2 seconds
80 km/h	1,8 seconds	3,6 seconds	5,5 seconds
100 km/h	1,4 seconds	2,9 seconds	4,3 seconds

Table 3.1: Time limits at different speeds and lengths from the tunnel

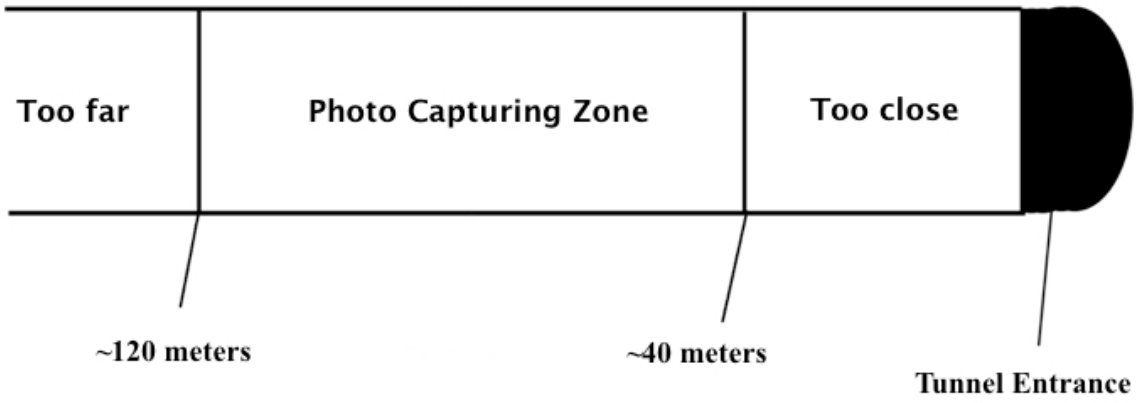

Figure 3.1: Distance classifications

3.2 Analysis of the road tunnel entrance

Describing the features of a road tunnel is a big challenge, because of the wide variety of shapes and lighting. In addition to this, road tunnels are built to blend as much as possible with nature [Staten06], which makes the tunnel entrance recognition a challenging task. In spite of these factors, humans do not have troubles identifying road tunnel entrances, and since we do not have trouble doing that, a computer should not have any problems either, in theory. For that reason this subchapter has been dedicated to describing what features humans see and how these features can be described in computational language.

3.2.1 Tunnel shape

Most tunnels are built from a cement construction. This cement construction can be in either a half-elliptic shape or a rectangular shape. The half-elliptic shape is the most common shape, especially outside of cities. The rectangular shape is used in cities, because the city tunnels often go under other roads and not through mountains, this means that the rectangular shape is easier to build and blends more with the “square” surrounding city. Image 3.1, Image 3.2, Image 3.3 and Image 3.4 shows examples of tunnel shapes.

Image 3.1: Half-elliptic tunnel

Image 3.2: Rectangular tunnel (city tunnel)

Image 3.3: Semi-circular tunnel (circular roof and straight walls)

Image 3.4: Rectangular tunnel with extended side wall and roof

3.2.2 The black hole

Thinking of the tunnel entrance as a black cave is an appropriate description. During the day, the entrance is often darker than the surroundings (Image 3.5), which is one of the dangerous properties of tunnels because human eyes are slow to adjust to the dark. This contrast is what the tunnel lights are there to reduce, but the opening is often darker than the surroundings nonetheless. This statement is correct during the day, but will slowly turn false when night falls. During the dark hours, the tunnel entrance will be a light source on the road (Image 3.6). Therefore, the “black hole” feature can only be used if the time of the day or the general illumination is taken into account.

Image 3.5: Daytime

Image 3.6: Nighttime

3.2.3 Tunnel ceiling lights

Tunnel lights are almost as diverse as tunnels entrances themselves. The number of lanes of lights in the ceiling ranges from one to four. Even the colors of the lights are different and can

be white, yellow or a mix of those two. In city tunnel the use of light in the upper corners of the roof is quite common which adds even more variety to the problem. Image 3.7 shows examples of the different light setups.

Image 3.7: Here you can see a single light line (left), double light lines (middle) and double corner light lines (right).

3.2.4 Obstructions

How do other obstructions affect these features? When driving toward a tunnel there is one obstruction that almost every time will present itself; namely other vehicles. A big brightly colored vehicle in between the car and the road tunnel entrance will make it difficult to spot the “black hole”. A big truck in the same position will cover the free sight to the tunnel lights and hinder the computer to locate the entrance. In these cases, humans don’t see the tunnel either. The only way for a system to see the tunnel entrance would then be if the truck were out of the way for one single picture. One way to increase the chance for this to happen would be to use two cameras on each side of the car. This way one of the camera would have a higher chance of detecting the tunnel entrance than a single camera would. This setup would also have a bigger chance of seeing the tunnel than the driver of the car would. Two cameras will however not be used in this experiment since the images are taken manually.

The problem of vehicles blocking the sight of the tunnel could also work in reverse. False tunnel identification could arise from having a big black vehicle in the middle of the picture. This is because of the similar features of the black vehicle and the black tunnel entrance. False tunnel identifications like this could also happen when there is a turn in the road and a black object is in the line of sight (in the middle of the image). This object can be anything from a dark forest to a dark building.

3.2.5 Localization

Knowing where the tunnel entrance should be located in an image can help the recognition methods in the initiation of the tests. Having the information of where to look in an image increases the likelihood of a correct classification. This information can also be used in another way: to test if the found object could be a tunnel entrance. Localization can therefore be useful in predefining initial variables or in validating hits on tunnels.

Tunnels are usually in the center of the images, both horizontally and vertically, which could be used to minimize the search space. There are however examples were the tunnel isn't in the center, which might make this information useless. Another way of locating the tunnel could be to find the road boundaries. These should then mark an outer frame for the search area.

One important detail is that the tunnel is placed upon the road. This information can be used as road boundary information by defining a search area and removing false hits on tunnels that are located on other roads than the vehicle is driving on.

3.3 The methods

When looking for methods that can recognize a road tunnel entrance, it is crucial to combine methods that capture the different features described in the previous chapter (3.2). Since the sought object does not have very clear features that separate it from other objects in the images, there will not be perfect results if only a single feature identifier is incorporated. For that reason a useful approach will have to include information of the road tunnel entrance that is as complete as possible. This chapter will describe different methods that can be useful in a hybrid system.

3.3.1 Tunnel Arch Detection

Generalized Hough transform was developed Richard Duda and Peter Hart in 1972 [Duda72]. The method is explained in more detail in chapter 2.3.1. The different versions of Tunnel Arch Detection (TAD) presented use a Circle Hough Transform function, implemented by Amin Sarafraz [Saraf04], to detect circles. Except for this function, the method is based on ideas and implementations created by the author.

The idea behind this method is to capture the shape of the tunnel entrance. The problem with using Circle Hough Transform (CHT) is that the classic tunnel shape is not fully circular, but more elliptic. In spite of this, the method might get some hits with a lower threshold (if it does not have to be more than one quarter of a circle correct). An example of this is shown in Image 3.8. Another limitation with the method is that tunnel entrances are half-circular (half-elliptic); this can be corrected by the use of an even lower threshold or by making changes in the Circle Hough Transform algorithm so that it only registers the upper half of the circle.

Image 3.8: The green line show circular hits, while the red show misses.

This method alone will not be able to identify or detect rectangular tunnel entrances (Image 3.9). The method is therefore restricted to only finding circular tunnel openings, which is not good enough for a complete system. However, if the method works well, it could be used in addition to another method that can detect other shapes.

Image 3.9: Here the whole circle is red because there is no detectable circular form.

3.3.1.1 Method designs

Three different versions of Tunnel Arch Detection have been developed:

1. TAD that uses CHT on a black and white image
2. TAD that uses CHT on an edge-image
3. TAD that uses a modified CHT on an edge-image that only registers upper half-circles

The shape of the road tunnel is the starting point in the process of finding a useful feature. The first method that is explained here uses Circle Hough Transform in a simplified way. Instead of using the CHT on an edge-image [chapter 2.2.3], it is used on a black and white image. Because the tunnel opening is darker than the surroundings in daylight and lighter in the night, the circular template will fit into that hole, defining the circular shape. With a set radius, the template will get multiple hits if the tunnel is close enough.

With no threshold for a needed amount of hits, the method got a hit percentage of 68 on the training set. This percentage is high mainly because of the low number of hits on images without tunnels (2 %). The method gave only a 38% correct result on images with tunnels. On

the test set, the method got a hit percentage of 52. Because of this, the method was rejected early.

Image 3.10: Tunnel arch hits with the first version of TAD

The second method that also is based on CHT use it in a more traditional way than the first method. Here the circles are detected in the edge-images.

After an edge detection-method produces the edge-image, the circles are localized in the edge-image using CHT. This produces a new image that contains information on every circular form in the tunnel image. Every circular form in this image with high enough value is identified as a tunnel entrance. A high value is achieved if the circle has a high amount of pixels divided by the circumference.

When setting the threshold value, for the number of pixels in a circle, no threshold excluded every false hit from correct hits. The tunnel hits are therefore not possible to separate from false hits with a threshold alone. The method needs a way to separate the hits with a higher degree of accuracy. One alternative is to remove unwanted noise around the tunnel (edges made by the surrounding environment). The approach would require a common feature for all the noise that does not affect the tunnel at all. This feature was not found and because of that, the method was rejected.

The other alternative way to separate correct hits from wrong hits is to calculate the noise within the circles. A tunnel have few edges (noise) within the dark opening, but a hit in a tree is bound to have a lot more edges inside its boundaries.

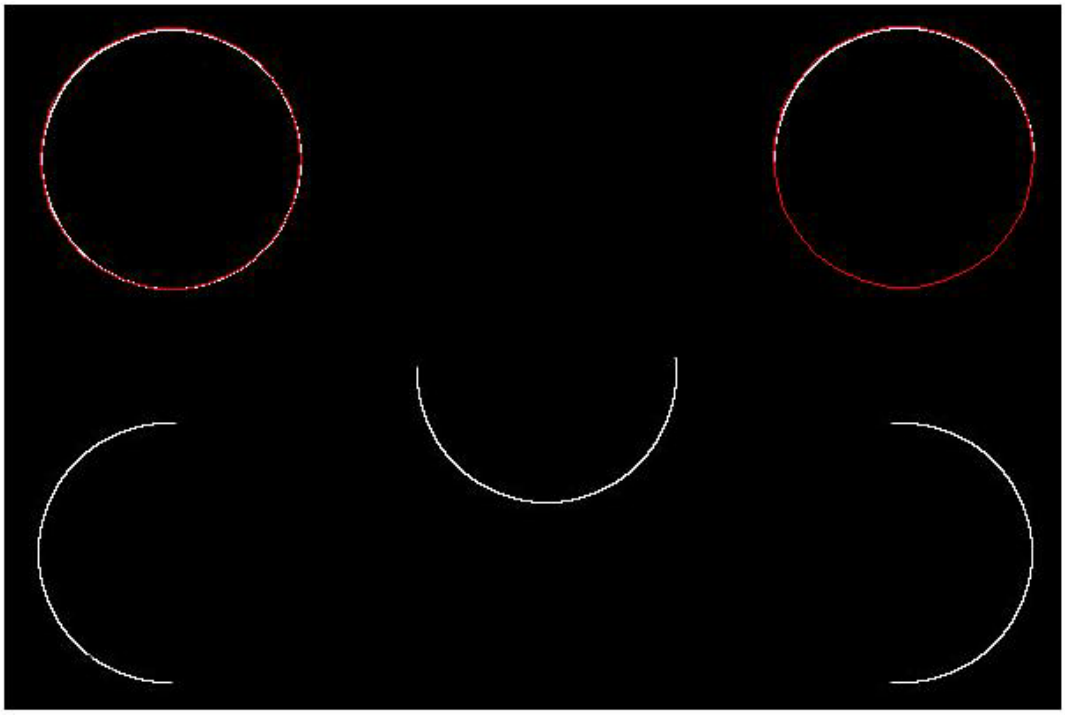

Image 3.11: One full circle and four half-circles to test the upper half-circle method.

The third version of TAD uses upper half-circles instead of whole circles but other than that, it is very similar to the second version of TAD. To detect half-circles CHT had to be adjusted so that it did not register the bottom half of circles. Image 3.11 displays results from the modified CHT. The method only detects the upper half of the circle, when the full-circle detection method would get hits on the other circle halves as well. Using half-circles instead of using a stricter threshold for the number of pixels in a circle, should give the method a higher change of "seeing" the half-elliptic tunnel entrance. However, it will also result in more detections since there are more half-circles in the environment than there are whole circles. Which version that works best is not as trivial as one would think.

3.3.1.2 Intermediate results

Multiple edge detection-methods had to be tested in order to maximize the result of this method. The reason for this is that the edge-image is an important part of the method, which

has a big influence on the result. The different edge detection methods are described in chapter 2.2.3.

From testing with the training set, Roberts' Cross produces the best result, as you can see in the Table 3.2. (This result has also been confirmed with the test set on both the whole circle method and the half-circle method.) This is a bit surprising because Roberts' Cross is a simple method that is often too sensitive of noise. Canny and Laplace has many correct classifications but total is ruined by a high number of false classifications. Thereby the choice fell on Roberts´ Cross.

Edge-methods	Tunnels classified (%)	No tunnel classified (%)	Total correct classifications (%)
Sobel	72%	38%	55%
Prewitt	72%	44%	58%
Canny	98%	12%	55%
Roberts' Cross	74%	60%	67%
Laplace	92%	20%	56%

Table 3.2: Values taken before adding noise calculation using the training set.

The amount of noise is calculated within the boundaries of the circle (or half-circle). It is calculated from the number of white pixels (edges) in the edge-image. Many different thresholds for how much noise that should be accepted have been tested to find the best one.

A variable threshold, like the radius of the tunnel entrance, will give a better result when working with multiple radii for the circle. The noise threshold had to be set according to percentage of noise (edges) inside the circle. The best circle threshold found was 90% of circle radius and the best half-circle threshold found was 70% of the circle radius. The best noise threshold was 7% for both methods.

Regarding the questions of which radii should be tested and how many different sizes there should be, the choice fell on eight different sizes ranging from 25 to 130 pixels. This range was found by calculating the tunnel radius in distance from 40 to 120 meters given in chapter 3.1.

Image 3.12: The result of TAD with half-circle detection. (Only half of circle is detected).

3.3.1.3 Final results

Results from third Tunnel Arch Detection method are listed in the three following tables (Table 3.3, Table 3.4 and Table 3.5). These are the results from detection of half-circles (the modified Circle Hough Transform). The whole circle method did not get as good a result with 64 % (training set) and 59 % (test set).

	Percentage Correct
Circular tunnel	44%
Square tunnel	20%
No tunnel	100%
Total	71%

Table 3.3: Half-circle results on the training set

	Percentage Correct
Circular tunnel	59%
Square tunnel	0%
No tunnel	98%
Total	71%

Table 3.4: Half-circle results on the test set

	Percentage Correct
Circular tunnel	47%
Square tunnel	0%
No tunnel	100%
Total	72%

Table 3.5: Half-circle results on the second test set

The results show that the half-circle version of the method has a more accurate classification than the whole circle version.

The every image set shows that the method will not detect square tunnel entrances, which were anticipated. The results have been adjusted to give very few false detections in non-tunnel images. This is important to get even higher results when using the method together with other methods in chapter 4.

3.3.2 Consecutive Tunnel Lights

The idea for the Consecutive Tunnel Lights (CTL) method came after multiple attempts to capture the darkness in the tunnel opening. Both the algorithm and the implementation are developed by the writer of the thesis. The attracting part of using a tunnel ceiling lights detection method is that it is not based on geometric shape detection as many of the other methods are. The ceiling light characteristic is not very consistent, however, because of the wide variety of setups and light types [chapter 3.2.3]. This means that the method can only search for bright lights next to each other and not be too specific with the arrangement of the lamps.

The idea is to first use segmentation to “highlight” the yellow/white objects in the picture. This will make it possible to access each object and evaluate if it belongs to a row of tunnel lights. The plan to do this is to search for yellow/white objects that stand in a vertical/diagonal line above each other and have individual properties as lamps have.

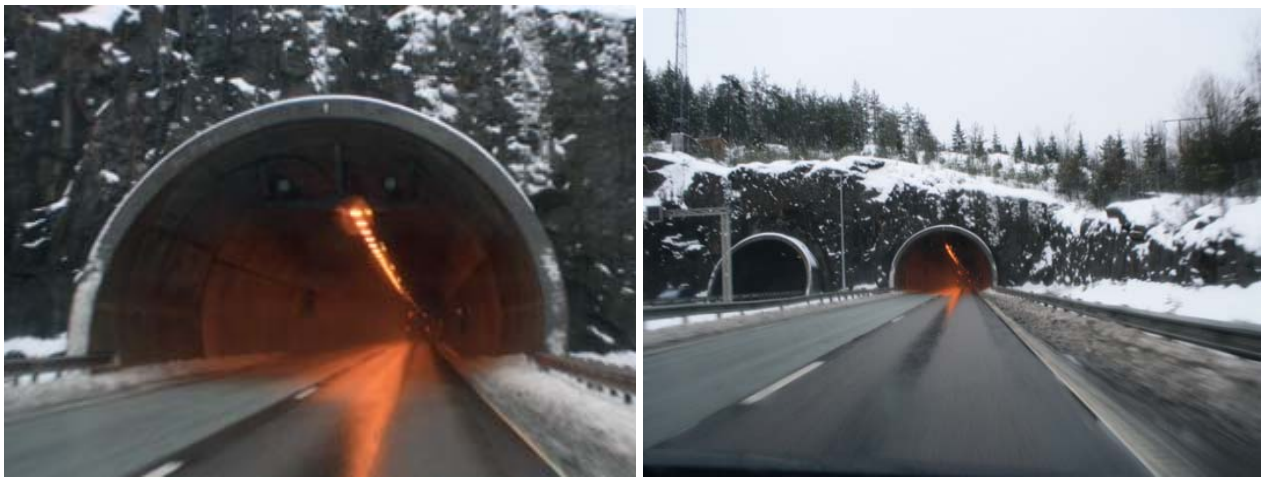

Image 3.13: 40 meters from tunnel (left) and 150 meters from tunnel (right). Both images are taken from sources with 600x400 resolutions.

Despite the difficulty that comes with the diversity of ceiling lighting, the biggest "roadblock" will lie in the detection of the lights from over 100 meters. This will be a challenge because of the small size of the lamps and because weather can minimize the visibility. In Image 3.13 you can clearly see that the visibility drop dramatically from 40 meters to 150 meters and makes the detection of individual light sources difficult. It is important to try to get correct classification at long distances also because of the time constraint in detecting the tunnel before entering it.

3.3.2.1 Method designs

Two different versions of Consecutive Tunnel Lights have been tested:

1. CTL filtered by color
2. CTL filtered by intensity values

In the first method, a RBG threshold [chapter 2.2.1] is used on the image with different rules for each color. The reason for doing this is to preserve the yellow lights that often are used in tunnels, together with the also used white lights. The thresholds for the different color components are displayed below. The values used are in Matlabs format with 0 representing black and 1 representing white. The rules were gathered by analyzing multiple images of tunnel ceiling lights. The constants used in the color filtering rules were found by checking pixel values in multiple images.

Color filtering rules for red (R), green (G) and blue (B):

1. Every (x,y) with R lower than 0.7 is removed.
2. Every (x,y) with G higher than R is removed.
3. Every (x,y) with G lower than R – 0.6 is removed.
4. Every (x,y) with B higher than G – 0.1 is removed.
5. Rules 1-4 are overridden if R > 0.9, G > 0.9 and B > 0.9.

Rule 5 is added to accept white lights as well as yellow lights. This does reduce the effect of the filter because it will not stop white objects like snow and clouds behind trees. The difference in results can be seen later in Table 3.6.

The second method does not make use of this color information. Instead, it works solely on a gray level image. From the gray level image an intensity threshold is calculated using Matlabs graythresh() function. The threshold is then used to turn the gray level image into a black and white image. From this point the two Consecutive Tunnel Lights methods are identical.

Bright objects are extracted from the segmented picture and marked with Matlabs bwlabel(). After this, regionprops() [chapter 2.4] extracts information from the objects for analysis. If the objects are real tunnel lights then the objects should make up a semi-straight line. If the method classifies an object as a light then other lights should be located below. A longer line of lights should be tunnel light objects and not car lights or some other bright objects. The longest line of objects in an image, within certain rules, is given as output from the method (Image 3.14).

Image 3.14: A successful registration

The rules for identifying the semi-straight line of lights are the important part of this method. The lights cannot be too far from each other or else they can be different kinds of objects, not necessary tunnel lights. The length between the lights will be longer if the tunnel is very close and short if the tunnel is far away. To make the variable length a dynamic variable it is linked to the size of the objects. The sizes of tunnel light objects are larger at close range and vice versa. With the length between the objects linked to the size of the current object, the following is valid: when the tunnel is close, the length between the object will be large, and when it is far away, the object will be small.

Figure 3.2: The search area for the next lamp.

Another important observation is that the next light in a line of lights (below the previous) is smaller than the other lights because of the perspective. This could used to remove hits that have an increase in object size instead of a decrease but because some lights furthest away merge together this would work against its purpose.

3.3.2.2 Results

From tests with both color filtering and intensity filtering, it is clear that the results of color filtering without rule 5 give a bigger difference between the longest lane of lights in tunnel images and the longest lane of lights in non-tunnel images (Table 3.6). The results using intensity filtering does not work acceptably with only 1.42 and 0.20 more lights in tunnel images than in non-tunnel images (these values are from the training set and the test set, respectively). Of the two color filter methods, the one without rule 5 give a better result than the one with rule 5. This is because rule 5 not only give room for white tunnel lights, it also give room for every other white object in the picture. In Image 3.15, you can see these wrong registrations that rule 5 adds.

	Average number of lights in Tunnel images	Average number of lights in Non Tunnel images
CTL with color filter	4.22 (4.76)	2.62 (3.28)
CTL with color filter (without rule 5)	3.18 (3.64)	1.20 (1.8)
CTL with intensity filter	5.90 (5.64)	4.48 (5.44)

Table 3.6: Average longest light lanes in the training set and the test set. The test set is in parentheses.

Image 3.15: Here white lines in the road are mistaken for lamps.

Full results from Consecutive Tunnel Lights filtered by color (without rule 5) are presented in Table 3.7, Table 3.8 and Table 3.9.

	Percentage Correct
Circular tunnel	87%
Square tunnel	80%
No tunnel	72%
Total	79%

Table 3.7: CTL results on the training set

	Percentage Correct
Circular tunnel	84%
Square tunnel	92%
No tunnel	40%
Total	62%

Table 3.8: CTL results on the test set

	Percentage Correct
Circular tunnel	61%
Square tunnel	0%
No tunnel	83%
Total	71%

Table 3.9: CTL results on the second test set

The results show that the method handles best circular tunnels. This is might be because circular tunnels have a single lane in the middle of the roof while square tunnels have different lighting setups. The differences between the image sets could be related to global lighting situations being affecting the algorithm. This will however not be tested in this report due to time limitations.

3.3.3 Lane Tracking

One important challenge when checking for a tunnel entrance is where to search in the image. This is important because searching through the whole image is time consuming and because the result when searching in a smaller area is more accurate. This problem manifested itself in every method that was tested. Tunnels were too frequently being detected beside the road and not on the road. This was partly because the search space was too big. With a smaller search space the results will be more accurate and the processing time faster. One way to do this is to only search the center of the image for a tunnel. This is the area tunnels usually appear. The problem with this approach is when the tunnel is not within these boundaries (Image 3.16).

Image 3.16: Off center tunnel

In some methods, it can be helpful to have the coordinates for a point in the tunnel, which can be used as a seed. A simple way of doing this would be to use the center of the image as a seed. This will meet the same problems as the example above but with even higher sensitivity to turns in the road before the tunnel.

A solution to this is to use Lane Tracking. This approach uses the knowledge that the tunnel entrance lays on the road and not beside the road or on a parallel road. By incorporating Lane Tracking, the road is registered and a smaller search space can be produced together with accurate coordinates to a point in the tunnel opening.

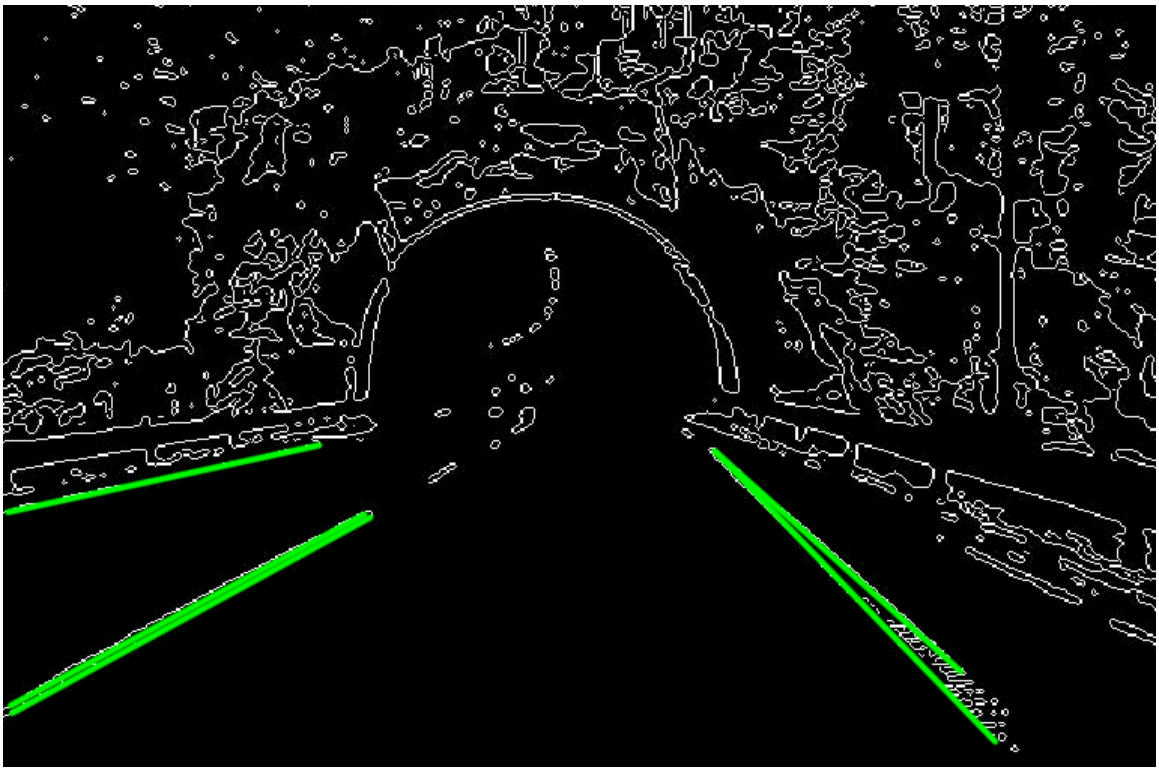

Image 3.17: The green lines are results from Lane Tracking

3.3.3.1 Method design

The implementation of the method uses multiple Hough Transform functions built into Matlab, other than that the method is developed and implemented by the author. The main process in this method is the Standard Hough Transform, which is used to find the major lines in the image. These lines are usually lane markings. This method however does not separate between lane markings and other major lines that indicate the direction the vehicle is driving. The lines are filtered through rules that make sure only the correct lines are identified. The lines always start in one of the lower halves (right or left) and end up closer to the center. After the filtering, all remaining lines are classified as direction lines.

The process to find the focus point start after the direction lines have been found. The focus point is calculated by finding every cross point between the lines on the left and the right. The mean of all these coordinates are then calculated to get the focus point. The focus point can now be used to start the search for a tunnel entrance since this point will mostly be in the middle of the tunnel opening.

Image 3.18: The yellow circle is calculated from Lane Tracking and the red square is the center of the image.

3.3.3.2 Results

In the test set, neither the center point nor the focus point gets good result. This is mainly because there are fewer pictures of highway tunnels in that group. Pictures of tunnels that have been taken in the mountains have very few lines that can tell the vehicles direction. Mountains and snow patches are registered as Hough lines, which make the estimation of where the tunnel is false. This is shown in Image 3.19 below.

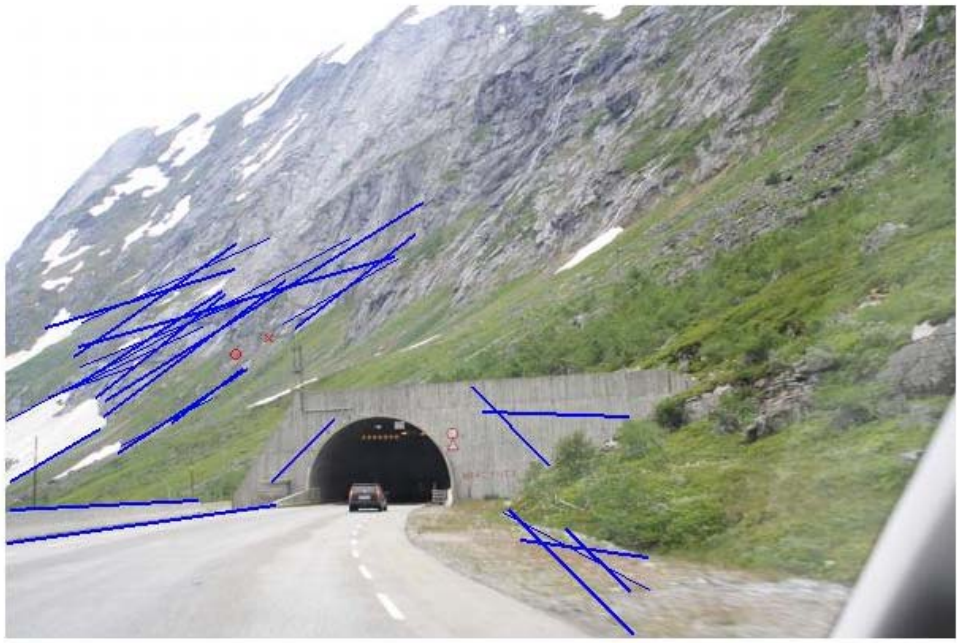

Image 3.19: Example of bad Lane Tracking in the mountains

	Center point in tunnels	Focus point in tunnels	Combined (optimal)
Training set	76 %	72 %	96 %
Test set	38 %	40 %	52 %
Test set 2	36 %	46 %	57 %

Table 3.10: Percentage correct of Lane Tracking seed and center seed.

The training set and the two test sets reveal that the use of Lane Tracking does not give a remarkable performance over the simple center seed method (Table 3.10). However, by using center seed together with Lane Tracking the result could improve with 11-24%. Many of the center points and focus points miss the tunnel entrance with a small distance. With extra seeds around the center point and the Lane Tracking focus point, the likelihood of a seed inside the tunnel entrance will increase. This has been done in the next method (Elliptic Black Hole).

3.3.4 Elliptic Black Hole (with Lane Tracking)

This method is meant to capture the tunnel feature that is most profound in daylight. At daytime people notices the tunnel entrance by the dark hole their driving toward [chapter 3.2.2]. This hole has the shape of the tunnel entrance that often is more elliptic than circular, thereby the name of the method. Lane Tracking is used with this method to improve the seed

needed for a better segmentation. Except for the region-growing function, both the idea behind the method and the implementation are developed by the author.

3.3.4.1 Method design

The method takes first a seed (coordinates) for where the tunnel entrance is estimated to be located. With these coordinates to a pixel inside the tunnel entrance, the whole tunnel is selected as a single object. Then the tunnel object is tested to see if it has the correct shape. The height and width of the object is used to create a perfect ellipse, which is divided horizontally and compared to the tunnel entrance. The amount of dark pixels compared to the amount of light pixels inside the ellipse should give a clue to if there is a tunnel in the image. This method thereby tests if the tunnel shape is correct and if the tunnel is darker than the rest of the image.

The Lane Tracking method was developed in the middle of the process of creating this method. The reason for that was this method's need for at good tunnel entrance locator. With the coordinates to a place inside the tunnel opening, the tunnel-object could easily be selected. Both Lane Tracking seeds and center seeds had to be tested to see if a good seed could be generated automatically. This ended in an algorithm where the Lane Tracking seeds are run first and followed by multiple center seeds. If the method finds a tunnel entrance with one seed than the method skips the rest of the seeds.

Two different segmentation methods were tested to capture the tunnel entrance object, threshold filtering and region-growing. A threshold is run during the tunnel entrance location and because of that, the method can reuse that data instead of creating new. The implementation of region-growing [chapter 2.2.2] is taken from an external source [Kroon08]. To find the best solution, both these segmentation methods have been run with either Lane Tracking seeds, the center seeds or a combination. Results show that threshold filtering with a combination of Lane Tracking seeds and the center seed. Region-growing with this combination were 11 % poorer than threshold filtering.

The objects in the image are given identification numbers and the object with the focus point pointing to it is selected as the candidate. The focus point is either a Lane Tracking seed or the center seed, depending on which is judged best by the algorithm. The height, width and

area of the candidate are then calculated with regionprops() in Matlab. With all this information, the ellipse, used to evaluate the object in question, is created.

The ellipse has the same width as the candidate but twice the height. The reason for this is to create a shape more similar to a tunnel entrance: a half-ellipse. The area of the ellipse is calculated and divided by two. The candidate is a tunnel if the area of the candidate is close to the area of the half-ellipse. A candidate that is not a tunnel entrance will most probably not have the same size area as the half-ellipse.

Image 3.20: Original photo before running EBH

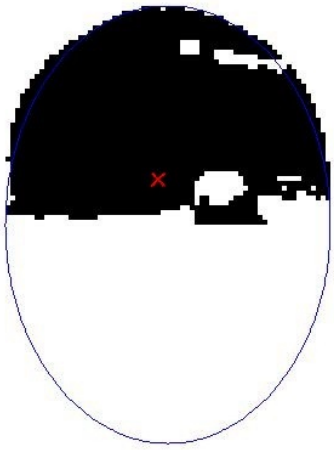

Image 3.21: EBH recognition of Image 3.20 (zoomed in)

Image 3.20 and Image 3.21 show what a recognition done by this method looks like. The red mark show the seed chosen by the algorithm and the elliptic circle show the shape that the black hole is compared with (the only the half of the ellipse is in the comparison).

The optimal ratio between the half-ellipse area and a tunnel object area is from 0.8 to 1.5. This range separates the unwanted objects from the tunnel objects. Only objects with area size between 980 pixels and 26530 pixels can be tunnel objects. This is calculated from tunnels entrance size at 120 meters and 40 meters, which were set in chapter 3.1.

3.3.4.2 Results

The results in Table 3.11 show that the method has a high percentage of correct classifications of unwanted objects. This means that the method seldom will give of a false warning to the

driver. The test set results for correct classifications in tunnel images not good, with only 40%, and means that the driver will be warned of under half of the tunnels he is approaching.

	Training set	Test set	Test set 2
Correct in tunnel images	96 %	40 %	64 %
Correct in non-tunnel images	96 %	96 %	90 %
Total correct	96 %	68 %	77 %

Table 3.11: Results from the EBH method with all image sets

3.3.5 Template Matching

Template Matching (TM) is often used to identify symbols in image analysis. These symbols are mainly numbers or letters but can also be special shapes like traffic signs [chapter 2.5.1] [Tørresen04]. The idea is to look for an object (template) in an image. A deeper explanation of how this works can be found in chapter 2.3.2. Using the Fourier domain to do template matching was developed by J.P. Lewis [Lewis95] and the following is the author's implementation of that method used in a tunnel entrance recognition system.

The reason this method works so well for symbols is that each individual symbol is closely identical to its twin and not other symbols. The details of one tunnel are however very different from that of another tunnel. The question is therefore if it is possible to simplify a template in a way that all tunnels can be identified with it.

3.3.5.1 Method design

To identify tunnels from multiple distances multiple sized templates is used. Using different templates for different shaped tunnels can increase the accuracy but it will also increase time consumption.

The crucial part of this method is the construction of the template. If the template does not resemble the sought object accurately then the method will not be of any value. There are two ways to build a template; the first is to build it from scratch. In the article [Tørresen04], the outer red circle on speed-limit signs was found using templates. Here a drawn generalization of the sought object was used as the template.

The second way to build a template is to use a sample of the sought object. This way works great if every object is very similar but not if there is a number of differences between them. Tunnel entrances are not similar enough to use this approach. Therefore, the choice fell on a more general method, namely building it from scratch.

Two different methods can be used if color information is an important part of the sought object. You can filter the images before template matching and keep only the interesting colors or you can use a template with the color information contained within. This template will have three color-layers (red, green and blue) that will have to be tested with the three color-layers of the main image. Here the first method will be used because it is easier to make fitting templates for it.

The Template Matching method filters important color information into two images: one to capture the tunnel entrance arch and the black hole, and one to capture the tunnel lights. This is to simplify the template images while including the must important information. The templates are black and white images of the "idea" of the circular tunnel entrance and its typical tunnel light setup (Image 3.22 and Image 3.23).

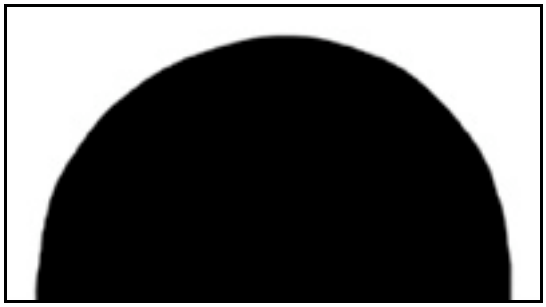

Image 3.22: The template for the tunnel arch

Image 3.23: The template for the tunnel lights

Fast Fourier Transform is used to find the frequency domains for the templates and the image. Then phase correlation is performed and the best match is found. The best match is then thresholded to separate tunnel matches from false matches.

3.3.5.2 Results

The classification threshold used during testing comes from maximizing the number of correct classifications in the training set. This resulted in a 69 % correct classification as seen in Table 3.12. In the test sets, the total correct classification was only 52 % and 53.5%. The method is clearly ineffective and because of this, the method will not be part of the RTERS.

	Training set	Test set	Test set 2
Correct in tunnel images	62 %	48 %	80 %
Correct in non-tunnel images	74 %	56 %	27 %
Total correct	69 %	52 %	53.5 %

Table 3.12: Results from the TM method with all image sets

3.4 Comparison of the methods

Each of the methods described in chapter 3.3 has its weaknesses and strengths. It is important to know these in a final system and if the methods are to be combined. For easier comparison of the methods, the results are repeated in Table 3.13.

	Training set	Test set	Test set 2
Tunnel Arch Detection (TAD)	71 %	71 %	72 %
Consecutive Tunnel Lights (CTL)	79 %	62 %	71 %
Elliptic Black Hole (EBH)	96 %	68 %	77 %
Template Matching (TM)	69 %	52 %	53.5 %

Table 3.13: Results from the main methods in chapter 3.3

EBH is a clear winner when looking solely on the training set results with 96% correct recognitions (17% higher than CTL which is placed at a clear second). EBH also comes first when looking at the test set result and the second test set result; here with an average 72.5% correct recognitions. All methods have poorer results with the first test set, except TAD, which show of very stable results. The poor result in the test set in some methods can mean that the methods were too specifically adjusted to fit the training set (this can happen if the image set is to small) but it can also mean that those methods works better in urban environments than in mountainous terrain.

When analyzing the resulting images that EBH produces with the first test set, there are indications that rocky hillside terrain make it difficult for the Lane Tracking method that EBH uses [chapter 3.3.3.2]. Since the simple Lane Tracking algorithm used here have troubles in this kind of landscape, one has to look at this sub-method to make it function better. The developed Lane Tracking method should therefore be replaced with a more functional one. This should not be a problem since in larger ITS systems a Lane Tracking method will be implemented to track the vehicles placement in the road. The method will then be used together with EBH, instead of this papers implemented version. This will also prevent that two similar methods run on the same system when one can do the job and lessen the load on the processor. In this report, however, the developed method, with its limitations, will have to be used.

3.4.1 Time consumption

There is another important side to the methods, other than the total correct classifications, which is important when evaluating the different methods, namely time efficiency. The time efficiency of the different methods show how many images the system can handle inside the Photo Capturing Zone [chapter 3.1]. If the vehicle is driving at 100 km/h the amount of time at disposal is only 2,9 seconds. For systems with lighter recognition algorithms like [Tørresen04] the system can run 24 times during 2,9 seconds. This is not the case for the methods developed for tunnel entrance recognition. In the table below (Table 3.14) the time efficiency for the different methods are shown.

	Average time consumption in the training set	Average time consumption in the test set	Average time consumption in the second test set	Total average time consumption
Tunnel Arch Detection (TAD)	0.9876 sec	0.9182 sec	0.9076 sec	0.9378 sec
Consecutive Tunnel Lights (CTL)	0.4014 sec	0.4034 sec	0.4341 sec	0.4129 sec
Elliptic Black Hole (EBH)	0.9688 sec	0.9750 sec	1.2704 sec	1.0714 sec
Template Matching (TM)	0.6378 sec	0.6451 sec	0.5974 sec	0.6267 sec

Table 3.14: Time consumption of the different methods used in this thesis

The numbers in Table 3.14 has been calculated from executions on a system with these specs:
Computer type: Apple MacBook Pro
Processor: 2.53 GHz Intel Core 2 Duo
Memory: 4 GB 1067 MHz DDR3
Operating System: Mac OS X version 10.6.1
Program: MATLAB® Version 7.8.0.347 (R2009a) 32-bit

The processing time for the method will be higher when implemented in a prototype since a small cheap processor will not have the same processing abilities as the MacBook Pro. However, running the algorithm in Matlab does not give the fastest run time either. CTL is clearly more time efficient than TAD and EBH. If only CTL was used in the system it would run 7 times in the Photo Capturing Zone, while TAD would run 3 times and EBH would run 2,7 times. For comparison, the time efficiency in [Clause06] is shown in Table 3.15, followed by specifications on the computer they used to calculate values.

1st Video	
Tunnel entrance detector	1.1496 s
Tunnel entrance detector 2	2.4816 s
Motion Estimator	4.7073 s
2nd Video	
Tunnel entrance detector	1,2768 s
Tunnel entrance detector 2	2,8030 s
Motion Estimator	5.4648 s

Table 3.15: Mean processing time measured for three main parts of the system measured from each video. [Clause06]

The system used in [Clause06]:
Processor: Intel® Pentium® 4 CPU 3.00 GHz , 2.99 GHz
Memory: 1,00 GB of RAM
Operating System: Microsoft Windows XP Version 5.1 (Build 2600: Service Pack 2)
Program: MATLAB® Version 7.1.0.246 (R14) Service Pack 3

Chapter 4

The hybrid system

This chapter introduces the next step after development and testing of each method independently. After seeing how the methods work on their own, you will now see how these methods can be combined into a final system. The challenge with this is to unite methods that cancel out each other's weaknesses and create a more complete program together. This new hybrid program has the potential to enhance the system accuracy toward 80-90 %. Keep in mind that this percentage is from image recognition on a single image and not image recognition on a series of sequential images. Because of this the results will not get as high as it would if the system also looked at a sequence of images. In future work this could be implemented.

4.1 Previous results and analysis

When evaluating which methods to use in the final system, it is important to look at both the results from each method and how the methods can complement each other. Making a hybrid of methods that does not focus on different features will not give the system extra useful information that a single method would not produce by itself. Therefore all this will be analyzed in this chapter.

When reevaluating the images in the training set and the test set, the images were more different than previously assumed. Since the training set is of tunnels and roads in the countryside and urban environment, while the test set comes mostly from a trip over the Norwegian mountains, there is a noticeable difference in the results with each image set. The rocky hillsides produce much more lines around the tunnel entrance compared with a flat environment. It is important that this is taken into account now, so methods that are not effective on mountainous terrain is not completely rejected. The optimal method should, however, be able to handle both urban environments and mountainous terrain. Because of this difference in image sets, a third image set was taken to make up for the weakness in the test set. This second test set has been run on each final version of methods, but has not been used in the development of the methods.

4.1.1 Analysis of hybrid alternatives

The methods described in chapter 3 that will be analyzed are Tunnel Arch Detection (TAD), Consecutive Tunnel Lights (CTL), Elliptic Black Hole (EBH) and Template Matching (TM). When combining some of these methods, it is valuable to look at how each method can be further tweaked to improve the result, but it is also critical to look at how the methods are combined.

The sequence in which the methods are combined will have a huge impact on the accuracy and speed of the system. The processing time for whole hybrid can be halved, depending on each methods run-time, if the first method that is executed in the hybrid can come to a definite conclusion early (a certain classification). If an early conclusion is unachievable then nothing can be gained in regard to the processing time.

The developed methods were made with two general ideas in mind. The first idea was to look at the difference in brightness between the environment around tunnels and the darkness inside the tunnel opening. The methods TAD and EBH use this as a base for analysis and detection. The second idea was to look at the light sources in the ceiling of the tunnel. CTL is based on this idea alone. TM is a combination of both of these ideas.

When choosing two methods to combine in the hybrid system, the classification presented in the paragraph above, gives information on which method could be combined with useful results. Methods that are based on two different general ideas will give more information to the whole system than two methods that come from the same idea. This means that the two interesting combinations to take notice of are CTL together with either TAD or EBH.

Since TM looks at both geometric features and tunnel lights, it should produce a complete result all by itself; therefore there is little need to look at a hybrid with TM. The only way this method can be relevant is if it produce better results with geometric recognition by itself or with tunnel light recognition by itself. This was, however, not the case with TM.

4.1.2 Simple combination rules

There are two ways to combine two results without merging the two methods into a specially designed method that may use the information in the methods better. Even though merging can result in a better outcome, a simple combination of the results can prove as functional if

the results from the separate methods complement each other. A simple combination can also be an indicator for what has to be done in the merged version.

M1 is the result from method number 1.
M2 is the result from method number 2.

And-rule: M1 and M2
Only images classified as tunnel images in *both* methods, gets classified as tunnel images in the combined result.

Or-rule: M1 or M2
Images classified as tunnel images in *either* method, gets classified as tunnel images in the combined result.

A third rule could have been established that would choose the most certain result. To do this the methods would have to return percentages instead of definite answers. This would however mean an adjustment to the methods codes and not be a simple combination rule. Method combinations like this are done in the two hybrids in chapter 4.2 and 4.3.

Each of the two simple combination rules work best in different situations. The And-rule works best when the methods have a high rate of false positives. The Or-rule is the opposite; it works best when the methods have a low rate of false positives. This means that the choice between the two rules is easily done by checking the results from both methods.

It is also possible to calculate the optimal effectiveness of the hybrid. If one method does a false classification while the other classifies it correctly, an optimal algorithm would choose the correct classification. However, if the two methods results overlap, i.e. gives no new information to the whole, then a combination of the methods would not produce better results. This estimated optimal result is referenced to as the upper boundary of the combined methods.

4.2 Hybrid of the EBH- and CTL-methods

Both the Elliptic Black Hole method and the Consecutive Tunnel Lights method gave reasonable results during testing. This alone makes them good candidates for a hybrid

program, but they also captures to different important features in the tunnel entrance. The Elliptic Black Hole method looks at the dark geometric shape of the tunnel opening and the Consecutive Tunnel Lights method looks at the bright objects within that dark geometric shape. Since the two methods do not overlap in their recognition approach, both methods have useful information to give the other method in its evaluation.

The upper boundaries of the combined methods are 100% for the training set, 95% for the test set and 89.5% for the second test set. It is not possible to combine the two methods and get a higher result than this.

When choosing one of the simple combination rules, the Or-rule is only barely better than the And-rule. The reason for this is that in EBH the non-tunnel hit percentage is better than the tunnel hit percentage and in CTL the non-tunnel hit percentage is worse than the tunnel hit percentage. This opposition makes none of the rules any good and the only way to improve this is to adjust EBH to be less critical in its judgments or CTL to be more critical.

Without adjustments the Or-rule produce a result of 84% in the training set, 65% in the test set and 79% in the second test set. To get a better result with the Or-rule, the low threshold of accepting all light lanes above 2 light sources has to be increased. With a threshold at a minimum of 4 light sources, the Or-rule results in 96% for the training set, 70% for the test set and 77% for the second test set. This is as high as EBH in the training set (96%) and the second test set (77%) but 2% higher than EBH in the test set (68%). This increase is not high enough to justify a 38% longer run-time compared to EBH.

Since the Or-rule was barely better than the And-rule in this case, an adjusted version of EBH was tested with the And-rule. This did not give as good results.

A closer integration between the methods could be done by comparing the coordinates from the classification performed in both methods and using that information to cancel out classifications that points to different parts of the image.

4.2.1 Hybrid 1

In the hybrid program, the two methods are merged in a way that maximizes the speed and the number of correct tunnel recognitions. An optimal sequence for the different image recognition algorithms creates a system that run as little code as necessary to find a solution.

Firstly a modified EBH is run to locate the black hole of the tunnel entrance and make analysis based on this feature alone. (The method has been modified to find the best black hole and not the first within a certain threshold.) If the difference between the object found and the elliptic circle is below 0.4 then the object is classified as a tunnel and the program ends.

Secondly a modified CTL finds the light objects in the image. (The method has been changed to return all light lanes and not only the longest.) If the number of lights in the longest lane is above 4 then the image is classified as a tunnel image and the program ends. This will incorporate square tunnels also, which might not have gotten as good results in EBH.

If the program is still running, it looks at light lanes within the black hole found by EBH. This is where the merging of the methods produces new information that is unavailable when the methods are separate. If there is a light lane inside the best black hole the image is classified as a tunnel image.

4.2.2 Results

The results show that the merged version of EBH and CTL has a higher correct classification percentage than the simple combination of the two methods (Table 4.1).

	Training set	Test set	Test set 2
EBH and CTL (simple combination)	96 %	68 %	77 %
EBH and CTL (Hybrid 1)	94 %	79 %	80 %

Table 4.1: Results from a simple combination of EBH and CTL and Hybrid 1.

The hybrid thereby get an average of 79.5 % for the test sets, which is 7 % higher than any method got individually in chapter 3.

4.3 Hybrid of the TAD- and CTL-methods

The Tunnel Arch Detection method did not produce as good results as the Elliptic Black Hole method but it was interesting because of its stability. The biggest variation in the EBH methods results was as high as 28% but the biggest variation in the TAD methods results was only 1%. This shows that TAD has the potential to produce more accurate results together with CTL than EBH had.

As described in chapter 4.1.1, TAD and CTL can be combined without overlapping any information about the tunnel entrance. TAD looks at the geometrical shape of the tunnel entrance and CTL looks at the lights within the tunnel entrance. With these two separate registered features, the hybrid program should improve the results when merging the two methods.

The upper boundaries of the combined methods are 94% for the training set, 93% for the test set and 90% for the second test set. This is, however, optimistic and one can only get to a hit percentage of this quality with a complex algorithm that merges the two methods completely.

The Or-rule is the best choice for combining TAD and CTL. This simple combination rule results in 83% accuracy in the training set, 66% accuracy in the test set and 77% in the second test set. This is 5.3% higher, in average, than the CTL methods results and 4% higher, in average, than the TAD methods results.

One way of merging TAD and CTL is compare the placement of the circular hits in TAD and the placement of the lane of lights from CTL, and use that information to remove false hits.

4.3.1 Hybrid 2

In the process of merging the two methods it was necessary to modify much of the code to make them compatible. In addition to changing the methods fit each other, the methods were changed to return information that was more useful in a combined analysis.

To get more information from CTL, the method was changed to not only return the longest light lane, but also other light lanes found by the method. This gives the main program much

more information to work with. The threshold for half-circles in TAD were lowered to get more information.

Another important part that was changed was the evaluation of the area around the lights. This was done in CTL to separate lights in the dark tunnel from the lights in the environment. However, in the hybrid it is better to do this check after all information from TAD and CTL has been collected. Thereby checking for the dark area around the lights within the half-circular tunnel opening and not just an arbitrary area around the lights. The whole light lane has to be within the half-circle to be accepted. This is done to remove light hits on objects other than tunnel lights. If no half-circular form is found around the lights, then the dark area is tested as if it is a rectangular tunnel opening.

The program analyze these features:
- Number of black pixels compared to area in the tunnel opening.
- Number of black pixels compared to area around light lane (when there are no half-circle hit around the lights).
- Highest half-circle result from the Tunnel Arch Detection.

These numbers are then thresholded and combined into the final result.

4.3.2 Results

The merged version of TAD and CTL is clearly more effective than the simple combination (Table 4.2).

	Training set	Test set	Test set 2
TAD and CTL (simply combination)	83 %	66 %	77 %
TAD and CTL (Hybrid 2)	83 %	76 %	87 %

Table 4.2: Results from a simple combination of TAD and CTL and Hybrid 2.

The reason for the high accuracy of 87 % in the second test set compared to the result in the training set can be located in the CTL methods results. In CTL test set 2 had fewer false classifications than the other image sets, and coupling this with the limitation the TAD circles gives, it produces the lowest false classification rate of all the image sets (3 %).

4.4 Comparison of the RTERS methods

Both hybrid versions of RTERS improved upon the results from the methods introduced in chapter 3.3. However, Hybrid 2 gives a better average result in the test sets than Hybrid 1. The entire average test set results from the methods described in chapter 3.3 and the two hybrid methods in this chapter are shown in the graph below (Figure 4.1).

The three methods, TAD, CTL and TM, are limited to detecting circular tunnel entrances. This reduces the correct classification percentage. The tunnel shape does not affect CTL as with the other methods. The hybrids are more effective than the single feature methods in chapter 3.3 because they use CTL, where tunnel shape is not an issue, together with a method that is more effective on the circular tunnel entrances.

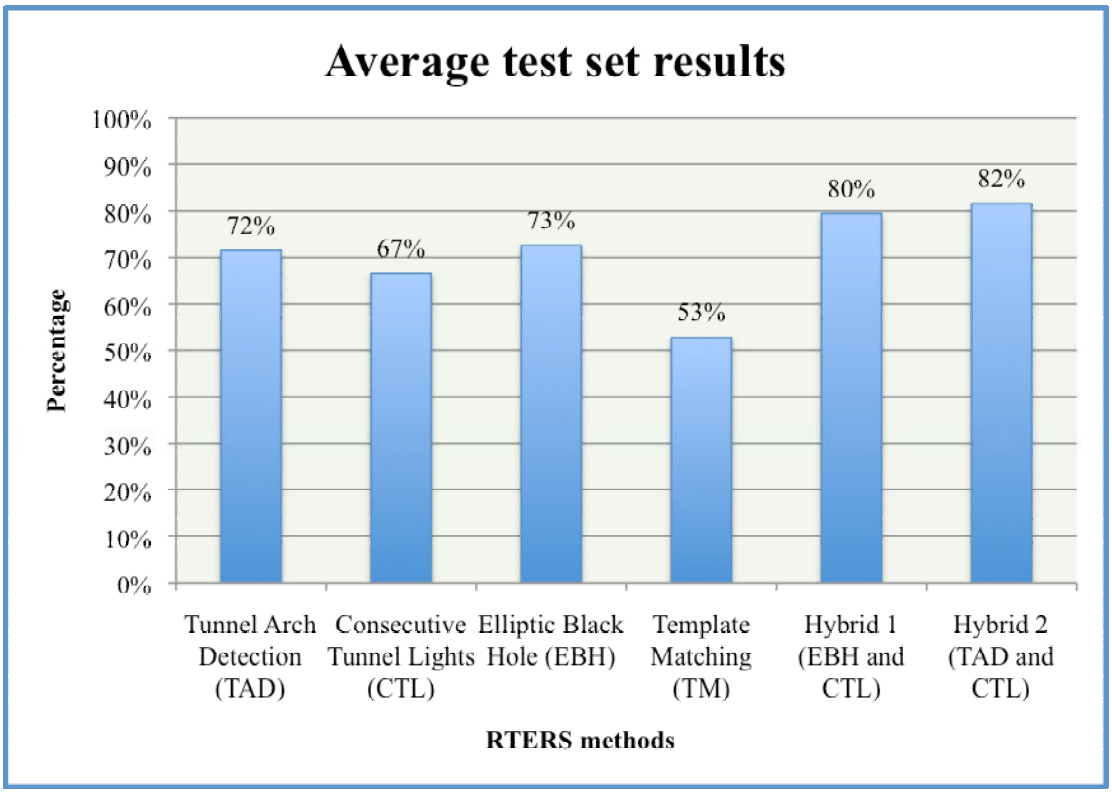

Figure 4.1: Average test set results

In addition to looking at the results, it is important to consider the methods time consumption, to check if the method comes to a conclusion before reaching the tunnel.

	Average time consumption in the training set	Average time consumption in the test set	Average time consumption in the second test set	Total average time consumption
Hybrid 1 (CTL and EBH)	1.3974 sec	1.4343 sec	1.7455 sec	1.5257 sec
Hybrid 2 (CTL and TAD)	1.4491 sec	1.3752 sec	1.4029 sec	1.4090 sec

Table 4.3: Average time consumption in Hybrid 1 and Hybrid 2.

From the information in Table 4.3, one can calculate that Hybrid 1 will run 1.9 times and Hybrid 2 will run 2 times, while in the Photo Capturing Zone [chapter 3.1]. This is pretty slow considering the hardware it is running on and if the prototype runs at less than half the speed of a 2.53 GHz Intel Core 2 Duo processor in Matlab, the system can miss the whole Photo Capturing Zone. This means that the RTERS does not get a chance to analyze the picture.

However, if the system can be optimized to run at the speed calculated here then the hybrids might get to analyze 2 images before reaching the tunnel. This would increase the likelihood of getting a correct classification above 82 %.

4.4.1 The optimal RTERS program

The evaluation of test results from each single RTERS method in chapter 3.3 and the hybrids in chapter 4, show that Hybrid 2 gives an optimal result.

The combination of Consecutive Tunnel Lights and Tunnel Arch Detection in Hybrid 2, gave a result of 82 %, which is an increase of 9 % compared to Tunnel Arch Detection used alone. The Hybrid 2 result is 2 % better than Hybrid 1 and is also 0.2 seconds faster (in the test sets).

The conclusion is that Tunnel Arch Detection, which use Half-Circle Hough Transform to detect the tunnel arch, combined with Consecutive Tunnel Lights, which detect tunnel lights, are the best method for finding tunnel entrances in still images.

Even though the results from tests on the selected methods show that Hybrid 2 gives the best result, there is room for further development of Hybrid 1. Because the Lane Tracking method is somewhat lacking in accuracy in rocky terrain, an improvement in that method could result in more accurate Hybrid 1.

The methods in this thesis have been analyzed and improved upon for a year, but there are still a potential for improvements in the described methods. Some of these improvements are listed in chapter 5.

Chapter 5

Further Work

Throughout this whole thesis one thing has made itself clear: There are several daylight and weather conditions which should be included in the image library and tested. In addition there are numerous tunnel opening designs and environment variations which have not been analyzed. For every angle to this topic that have been studied there have been unveiled several alternatives which needs to be checked. Each alternative should be tested to see if it could be included in a Road Tunnel Entrance Recognition System.

This thesis has been developed from scratch on an idea that the RTERS could be based on programs that recognize forms and shapes captured in photo images taken by a camera in the front of a vehicle. Step by step the system has been formed by not only the positive test results after several ideas have been studied but also the negative results. Each time an idea have been tested and the results have been evaluated new leads have been discovered. Most have led into dead ends but some have given results which now are included in this thesis and described in previous chapters.

The system introduced in this thesis still include many interesting challenges for those who would like to fine-tune the system and make it more accurate with existing programs, by adding new programs or combining existing ideas with new. The system could include several other functions when it comes to recognizing tunnel entrances. There are some topics mentioned below that might improve the functionality of the system.

5.1 Use of video-stream

This subject will have a large influence on the effectiveness of the Road Tunnel Entrance Recognition System. The transition from single image analysis to the analysis of a series of images would give the system much new information to interpret.

In the developed system using single images you get only 2 dimensional information to judge if a image has a tunnel entrance or not. This is a simplification of how it would be in the implemented system. In the implemented system you would get a continuous stream of images that could be analyzed as 3 dimensional information, with the third dimension being

time. If one image is classified as a tunnel image, then the next image could be used to verify that classification.

After finding a tunnel candidate in one image a lot of new information is available. This information can be used to get a better analysis of the tunnel in the next image in addition to minimizing the needed processing time.

Firstly the RTERS would know where in the next image to look for the tunnel entrance, thereby canceling out the need for a method to search the whole image to find the tunnel entrance center. This will save time and result in an accurate estimation for a small search area. In EBH this means that the Lane Tracking algorithm can be skipped in the second images because the tunnel entrance coordinates are already know.

Secondly you would know the shape of the tunnel entrance in the next image. With that information you also know the size of the tunnel you are looking for by estimating in how fast you are approaching the opening. This will for example help TAD in that it can adjust the circle radius automatically so it do not have to do any unnecessary searches.

Also, when analyzing if the tunnel in a series of images is correctly identified, the size and shape can be used to confirm that it is indeed the same tunnel entrance.

Other recognition systems, such as Road Sign Recognition, will likely use video-stream from a camera in the front of a vehicle as well. A combination of systems will be easier if RTERS use the same input format as other systems. The use of a video-stream should therefore be prioritized in further research.

5.2 Tunnel recognition at night

Recognizing tunnel entrances in daylight is an important part of a tunnel entrance recognition system but it is also crucial that the system is equally effective at night (Image 5.1).

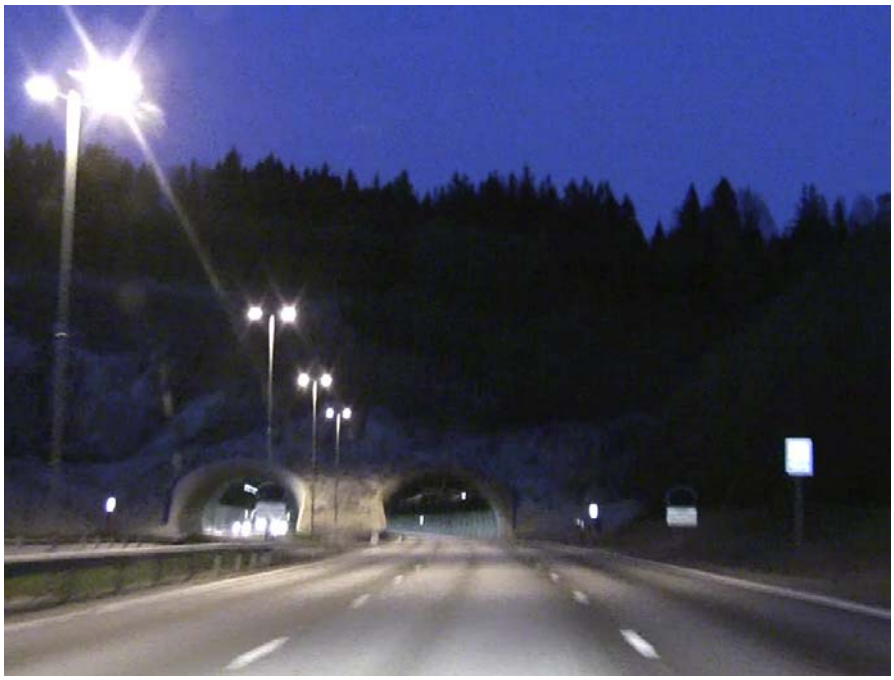

Image 5.1: A tunnel image taken at night

The driver will not be as blinded when driving into a tunnel at night because the light difference is less, but recirculation of air is just as important at night. The toxic atmosphere could also easier affect the driver when it is in combination with potential sleepiness.

A system for recognition of tunnel entrances at night will have to use different methods than the ones that work in daylight. At night the tunnel entrance is lit up and the surroundings are dark. This means that the methods cannot rely on the darkness of the tunnel opening to find it. It will have to use the opposite brightness of the tunnel opening. This could be fixed by simply “inverting” the methods for daytime but will have to be thoroughly tested.

The illumination of the road towards the tunnel and at entrance will be of several different types and intensities. In some cases the darkness outside the road – and above, will give an impression that the vehicle are in a tunnel. This situation has not been evaluated and tested as part of this work. How this situation should be interpreted by the RTERS will be one of the challenges that need to be solved in the next version of RTERS.

5.3 Tunnel recognition at dusk and dawn

The reduced intensity of daylight between day and night; i.e. at dusk and dawn, will also represent a challenge for the Road Tunnel Entrance Recognition System. The contrast in photo or video-stream images will not be as sharp as in daylight. The contrast to the darkness of a tunnel opening during daytime has been one of the main properties searched after by the RTERS. This will not be as obvious at dusk and dawn. There will be a need to find other properties which the RTERS should search for.

The transition from day to night and back will have to be measured to find the appropriate period for daylight tunnel recognition and nighttime tunnel recognition. This swap between these modes could either be done with a clock or a light sensor.

5.4 Different types of weather

Rain and fog are huge problems when it comes to visibility. For the system to be operational during these types of weather, it needs to be tested thoroughly. When the line of sight is reduced because of weather, the need for accurate information increase and safety of the driver depends on correct information from the RTERS. It is helpful with all additional information a camera can give the car to increase traffic safety, as long at this information is correct.

In addition to reduce the line of sight, poor weather also reduce the road grip, which means the driver has to react even faster that in good weather.

Rain will reduce the reflection from the road and other surfaces that are not in an 90 degree angle to the driving direction of the vehicle. All these surfaces will turn black. In contrast, the reflecting lines on the road will be highly visible, as will road signs and other reflecting objects that have been installed to guide the driver safely through the road system. Some of these reflecting objects could be so bright that they would dazzle the camera as if an approaching vehicle should keep the headlights high.

Raindrops on the camera lens would also reduce the quality of images which the RTERS tries to interpret. Such disturbances should be included on the list of problems to be solved for the next RTERS version.

The same reduced contrasts as with dusk and dawn, will happen when the weather changes to fog and the road lights are both dimmed and reflected in the hovering drops of water that forms the fog. The light from posts, other cars, buildings on the roadside and the tunnel illumination will be reflecting in the fog in a way that shades the visibility to the road tunnel entrance. The problem will increase with the intensity of the fog.

5.5 Difference in seasons

Seasonal changes have not been tested as part of this thesis. Each season come with a different range of colors and vegetation that will have influence on the algorithms.

Image 5.2: The tunnel in Image 5.3 photographed in November

Image 5.3: The tunnel in Image 5.2 photographed in February

In spring trees go from leafless to fully green. A leafless tree by the road can trick the CTL method to think it is a tunnel with its complex texture. This was a challenge when developing CTL and there are probably other unknown challenges with this season.

Summer comes with bright colors and might be the easiest season to recognize a tunnel entrance in daylight. Even during summer night the visibility should be better than other seasons due to the fact that most dominating colors (shades of green) are still not reflecting much light from road illumination or vehicle headlights.

Autumn can come with some challenges however. Colors are darker and leafs can fall onto the road hiding the road markings. Even the colors of the autumn leaf will be a challenge to sort out from road tunnel illuminations, as the colors are in many cases very similar.

Winter produces the biggest difference in the environment, namely snow. Snow covers much of the road and sides of the road. In some images used in this thesis snow has produced lines that confuse the recognition algorithms.

During winter some countries have snow covering most of the ground outside the road and round the road tunnel entrance. The contrast between ground covered by snow and areas not covered represent a challenge for the RTERS, in that it will have to separate these shapes from the road tunnel entrances.

5.6 Different tunnel entrance designs

During the research done on tunnel entrance design it has been stated that there are several alternative shapes in addition to the half circular. These shapes should be more thoroughly included in the criteria for the RTERS. Some entrances, mainly in urban areas, are rectangular. Such tunnel entrances are placed under road bridges or city center buildings. Tunnels that lead the traffic under rivers or artificial waterways such as ducts and channels have the entrance at the end of funnel-formed road. How the RTERS handle the different tunnel shape has to be tested further.

The RTERS should also include functions that separate road sections that pass under bridges from roads that pass through tunnels.

5.7 Driving in a tunnel and leaving it

When the RTERS activates functions that protect the passengers in the vehicles from the toxic atmosphere in a tunnel, it should also be able to detect when the vehicle leaves the tunnel. The system should then be able to switch back all function to the status they had before the vehicle entered the tunnel.

For further tunnel safety, the system could also be programmed to identify escape doors, first aid stations, emergency phones and more, when in the tunnel. This information could then be automatically shown on a screen in the vehicle if it by some reason should stops in the tunnel.

5.8 Sunlight

The direction of sunlight either facing the vehicle or any other angle to the vehicle would also have an effect on the image which the system shall analyze. The shadows will have a great influence on how the tunnel entrance appears. If the sun or another source of light appears in front of the vehicle, it can cause flares or otherwise disrupt the image processing. These light conditions will have to be tested as well.

5.9 Implementation in a prototype

The final part that has to be done to complete this work is to build a prototype. With a prototype, it is possible to perform real-time testing while driving and new unconsidered problems can be detected.

To build a prototype the algorithms has to be put on a portable processing system. This would probably be on a system-on-a-chip (SoC) with a FPGA (field-programmable gate array). A FPGA is a component with configurable logic functions, which is possible to repeatedly reconfigure after it has been produced. The reason to use a FPGA, instead of only running the system on the processor on the SoC, is to increase the speed of the process. Image operations are 8 to 100 times faster on a FPGA compared to an 800 MHz Pentium 3 processor [Draper03].

The camera used should deliver a video feed with the resolution 600x450 pixels, which is the resolution used in this thesis. With a higher resolution the classification might be more accurate but the loss in processing speed will be significant. If a lower resolution is wanted to increase the speed then the optimization will have to be redone, and the performance of the system cannot be guarantee.

Chapter 6

Conclusions

6.1 Summary

This thesis is an evaluation of different image recognition methods which could be used as a ITS to automatically prepare a vehicle for a safer drive through a tunnel. The system has been named Road Tunnel Entrance Recognition System (RTERS).

As part of the preparations for this work, 3 series of photo images were taken through the front window of a vehicle, when it approached different tunnel entrances in the city of Oslo, on the highways through the countryside and on the roads crossing the high mountains of Norway. The three photo series consisted of 100, 100 and 200 images. In each series, half of the images are taken within a distance of approximate 120 to 40 meter from a tunnel entrance. The other half of the images is of roads without any tunnel entrance.

The goal of this thesis has been to select methods that separately or in combinations could give a stable and reliable system for the recognition of tunnel entrances. The methods have been tested on the three image series to see how effective they are in recognizing different types of entrances.

The tunnel entrance recognition methods tested in this thesis are:

Tunnel Arch Detection [chapter 3.3.1]:
The half-circular arch is a common design in tunnel entrances and the idea behind this method was to capture that shape. To capture this arch, a Half-Circle Hough Transform function was used. The function is based upon the Standard Hough Transform [chapter 2.3.1] and detects every upper half-circle in an edge-image [chapter 2.2.3] created from the input image. These detections are then analyzed and if one meets the criteria in solidity and has little noise within its boundary, the image is classified as a tunnel entrance image.

Consecutive Tunnel Lights [chapter 3.3.2]:
Most tunnels use strong lights in the entrance to lessen the transition from daylight to the darkness in the tunnel. The lights are often visible from a long distance and Consecutive

Tunnel Lights are an attempted to detect a lane of tunnel lights. This is done by color filtering the image to extract the tunnel lights and then these lights are analyzed to see if they are part of a lane of lights. If a long lane of lights is found then it can be assumed that the image has a tunnel entrance.

Elliptic Black Hole [chapter 3.3.4]:
The black hole is a description of the tunnel opening during daytime because the tunnel opening is usually very dark except for the small light sources in the ceiling. The Elliptic Black Hole method thresholds the image to remove bright objects and check the remaining objects if they have the elliptic property of an elliptic tunnel entrance. Lane Tracking is used to help in choosing which objects to evaluate.

Template Matching [chapter 3.3.5]:
By generalizing the tunnel arch and the tunnel lights into two templates, Template Matching should be able to find matches for the templates in the images. When creating the templates, many circular tunnel entrances had to be analyzed to find the most general version possible of a tunnel arch and the tunnel lights. The Template Matching is preformed with Fast Fourier Transform, which finds matches in the frequency domain [chapter 2.3.2].

The average results from the methods are shown in Table 6.1.

	Average test set result
Tunnel Arch Detection	71.5 %
Consecutive Tunnel Lights	66.5 %
Elliptic Black Hole	72.5 %
Template Matching	52.8 %

Table 6.1: Average results from the test sets [chapter 3.4].

To improve upon the result given in these methods, two hybrid methods were developed. The hybrids were made to combine tunnel entrance recognition methods that were based on different feature detections.

Hybrid 1 [chapter 4.2]:
This hybrid is a combination of the two methods, Elliptic Black Hole and Consecutive Tunnel Lights. The Elliptic Black Hole method detects the geometric shape of the dark tunnel entrance and the Consecutive Tunnel Lights method detects the bright objects located within the dark tunnel entrance. By combining these two different feature detection methods, the result should be an improvement over both methods by themselves.

Hybrid 2 [chapter 4.3]:
Tunnel Arch Detection and Consecutive Tunnel Lights are two methods that could be combined to enhance the RTERS, in addition to the combination in Hybrid 1. Here Tunnel Arch Detection finds all half-circles in the image, and the localization of these are compared with the localization of the lane lights found by Consecutive Tunnel Lights. If the method finds a good half-circle with a long light lane within then the image is classified as a tunnel image.

The average results from the hybrids are shown in Table 6.2.

	Average test set result
Hybrid 1	79.5 %
Hybrid 2	81.5 %

Table 6.2: Average results from the test sets [chapter 4.4].

6.2 Conclusion

In addition to a high accuracy in its classifications, the RTERS has to be able to evaluate as many images as possible in a short period of time. This is foremost to make sure that the system evaluates a image taken between 120 and 40 meters before the tunnel, which the tunnel entrance classification is based upon [chapter 3.1], but so that a expanded RTERS could evaluate the sequence of tunnel images. The average time consumption is shown in Table 6.3.

	Average time consumption in the test sets
Tunnel Arch Detection	0.9129 sec
Consecutive Tunnel Lights	0.4188 sec
Elliptic Black Hole	1.1227 sec
Template Matching	0.6213 sec
Hybrid 1	1.5899 sec
Hybrid 2	1.3891 sec

Table 6.3: Average time consumption in the test sets [chapter 3.4] and [chapter 4.4].

Every developed method has shown some ability to recognize a tunnel entrance and has helped in process of perfecting the RTERS. The method that was finally chosen for the RTERS program was Hybrid 2, which was 2 % more accurate than Hybrid 1 and 0.2 seconds faster.

Even though Hybrid 2 has produced a relatively good result, there is still a potential for further improvements. This could be done by incorporating video-stream analysis instead of still image analysis, adding methods that can detect road tunnel entrances at night and upgrading the methods so that they work in all kinds of weather.

Index of Figures, Images, Tables and Equations

Figures

Images

Tables

Equations

Appendix

CD information

The program code for every method is included on the CD that comes with this report. The content are as follows:

The Methods folder:

TAD3.m	- Tunnel Arch Detection 3
CTL.m	- Consecutive Tunnel Lights
EBH.m	- Elliptic Black Hole
TM.m	- Template Matching
colorFilter.m	- Filter function for CTL.m
lightFilter.m	- Filter function for TM.m
houghcircle2.m	- Half-Circle Hough Transform
houghtrans.m	- Linear Hough Transform
Archi200.jpg	- Arch template for TM.m
Archi250.jpg	- Arch template for TM.m
Archi300.jpg	- Arch template for TM.m
Archi350.jpg	- Arch template for TM.m
Light200.jpg	- Light template for TM.m
Light250.jpg	- Light template for TM.m
Light300.jpg	- Light template for TM.m
Light350.jpg	- Light template for TM.m

The Hybrids folder:

RTERS1.m	- Hybrid 1 using CTL and EBH
RTERS2.m	- Hybrid 2 using CTL and TAD
RTERS_lights.m	- Modified CTL for use in hybrids
RTERS_EBH.m	- Modified EBH for use in Hybrid 1
RTERS_TAD3.m	- Modified TAD3 for use in Hybrid 2
colorFilter.m	- Filter function for RTERS_lights.m

Additional Method Versions folder:

TAD1.m	- Tunnel Arch Detection 1
TAD2.m	- Tunnel Arch Detection 2
houghcircle.m	- Circle Hough Transform by Amin Sarafraz
regiongrowing.m	- Region-growing by D. Kroon

Bibliography

[Statens06] Statens vegvesen, Håndbok 0121, *Vegtunneler,* 2006

[Transp00] Transportøkonomisk Institutt, *Trafikksikkerhetshåndboken*, *1.19 Sikring av tunneler,* 2000

[Morawska09] Morawska L., *Air pollution in tunnels,* Queensland University of Technology, International Laboratory for Air Quality and Health, 2009

[Clause06] Clause C., Shin H.C. and Stechele W., *Tunnel Entrance Recognition for video-based Driver Assistance Systems,* Technical University Munich, 2006

[Tørresen04] Tørresen J., Bakke J. W. and Sekanina L., *Efficient Recognition of Speed Limit Signs*, University of Oslo, 2004

[Jamal] Jamal H., Sami-ud-din and Habib H. A., *Road boundary detection in night video sequence: A novel technique for autonomous vehicle.*

[Nagumo] Nagumo S., Hasegawa H. and Okamoto N., *Extraction of forward vehicles by front-mounted camera using brightness information.*

[Marko] Marko K. A. and Hampo R. J., *Application of genetic programming to control of vehicle systems.*

[Gonzalez92] Gonzalez R. C. and Woods R. E., *Digital Image Processing*, Chapter 6.7.2. Pearson Education, Inc 2008

[Foley97] Foley and Van Dam, *Computer Graphics*, Lecture foils, Chapter 13, 1997

[Solberg08] Solberg A., *Shape descriptors*, Lecture foil in INF4300, 1. Oct. 2008

[WikiSobel] http://en.wikipedia.org/wiki/Sobel_operator, 15. Nov. 2009

[WikiPrewitt] http://en.wikipedia.org/wiki/Prewitt, 15. Nov. 2009

[MatEdge] http://www.mathworks.com/access/helpdesk/help/toolbox/images/edge.html, 15. Nov. 2009

[houghp] http://moscoso.org/pub/video/opencv/svn/opencvlibrary/trunk/opencv/doc/ref/pics/houghp.png, 15. Nov. 2009

[Duda72] Duda, R. O. and P. E. Hart, *Use of the Hough Transformation to Detect Lines and Curves in Pictures*, Comm. ACM, Vol. 15, pp. 11–15, Jan. 1972

[Saraf04] Sarafraz A., http://www.mathworks.com/matlabcentral/fileexchange/4985-circle-detection-via-standard-hough-transform, University of Miami, 29. Nov. 2004

[Kroon08] Kroon D., http://www.mathworks.de/matlabcentral/fileexchange/19084, University of Twente, 6. Mar. 2008

[Lewis95] Lewis J.P., *Fast template matching*, Industrial Light & Magic, 15. May 1995

[Draper03] Draper B. A., Beveridge J. R., Böhmand A. P. W., Ross C. and Chawathe M., *Accelerated image processing on FPGAs*, IEEE transactions on image processing, 12(12):1543-1551, Dec. 2003